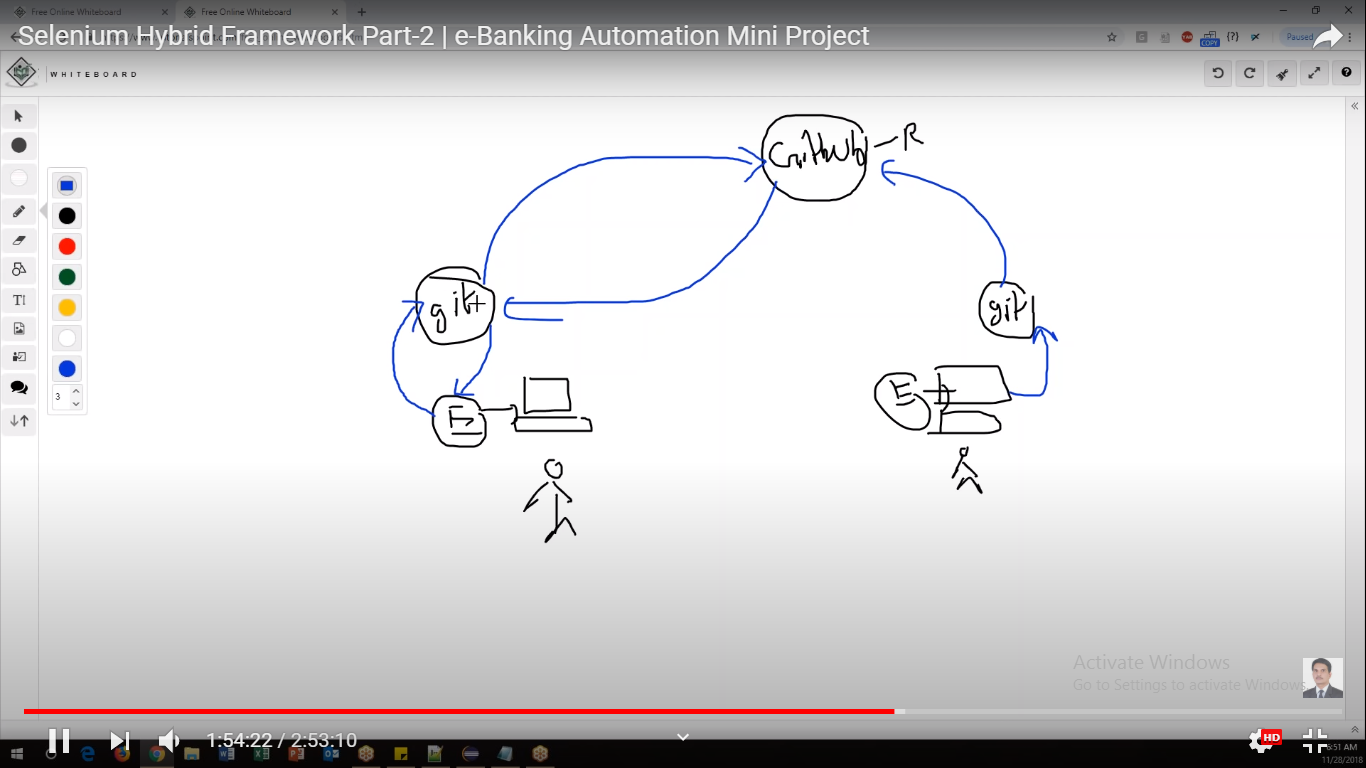
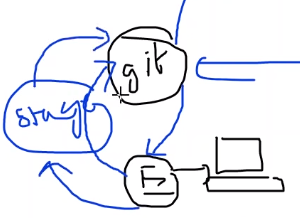
* What is continuous integration process?
* In a team of software development there are various teams like Dev team, Devops team, Testing team.
* Developers will develop the codes, tester will test the build and Devops will facilitate the build in the form of .exe file.
* Devops team will create the builds automatically using build tools. E.g. Maven is a build tool. It is used to create the build.
* Developers will develop the codes and keep it in a repository. From there Devops will create a build.
* Likewise QA team will also write their automation scripts an keep them in a repository.
* Devops will pick the automation codes to their environment and run it using Jenkins. Sanity and Smoke test cases they will run.
* Once Automation smoke/sanity gets passed, mail will be triggered to the configured contacts. It’s Jenkins functionality.
* QA will always download the build from Jenkins itself.
* Once Dev/QA members are done with coding for the day, they will keep the codes in repository. Throughout night Devops will create builds and mails if the build is success/failure. Next day morning we can download the builds and test/develop them again. It’s a continuous process. It is known as Continuous Integration Process.
* Jenkins will be installed in Devops environment.
* This Repository concept is a maintenance part.
* Git and GitHub
* Git is a local repository and github is a remote repository.
* Git we have to install in our local system.
* Codes are saved in local inside Workspace folder.
* Once any changes done in local, we have to commit those changes in git.
* Once any changes done in local, we need to commit those changes in git repository.
* After that we have to send them in github repository.
* Suppose someone else have done some changes in github repository, we can get them in our local also. It is a two way communication.



* There are certain commands to do that.
* Between git and local there is a buffer area which is known as Staging area.

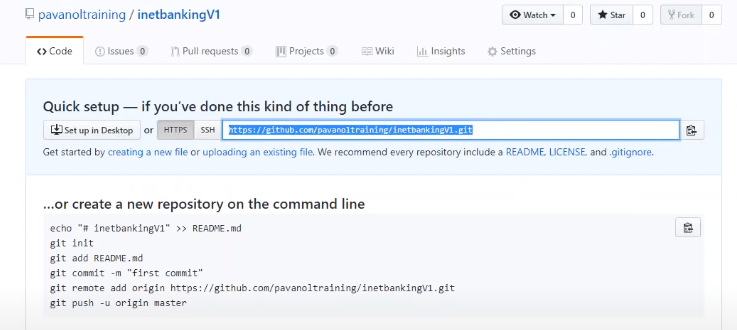


* The changes done from git to github are known as check in process. Reverse is known as check out process.
* Local repository: git
* Remote repository/global repository: github
* Prerequisite:

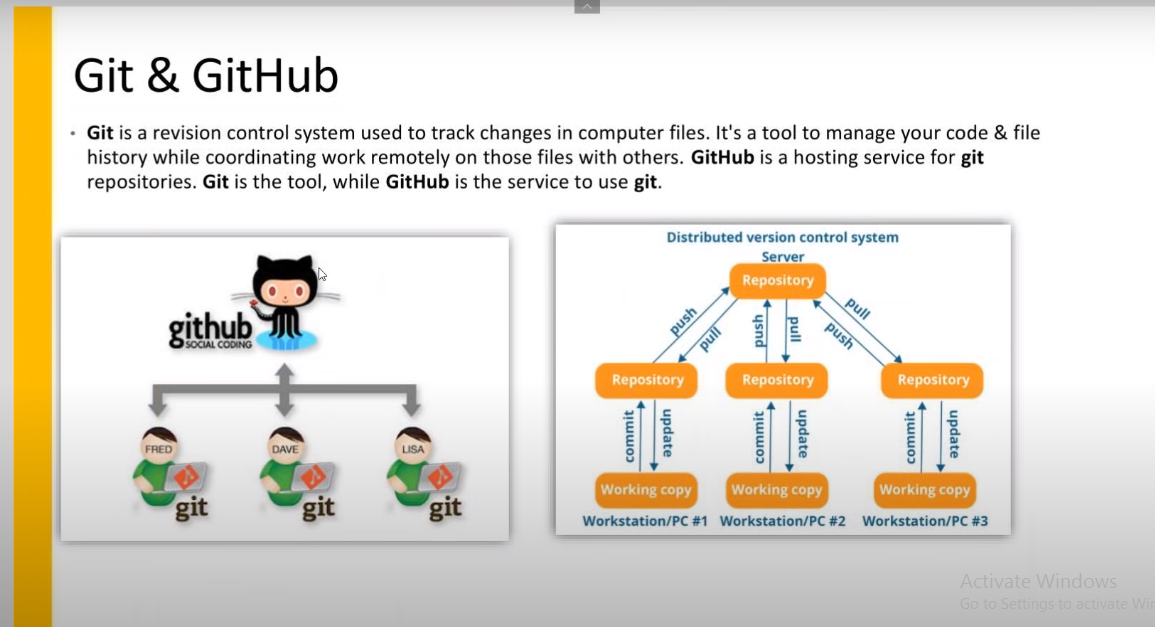
1. Install git in local system.
2. Create Remote Repository in github.

<https://github.com>

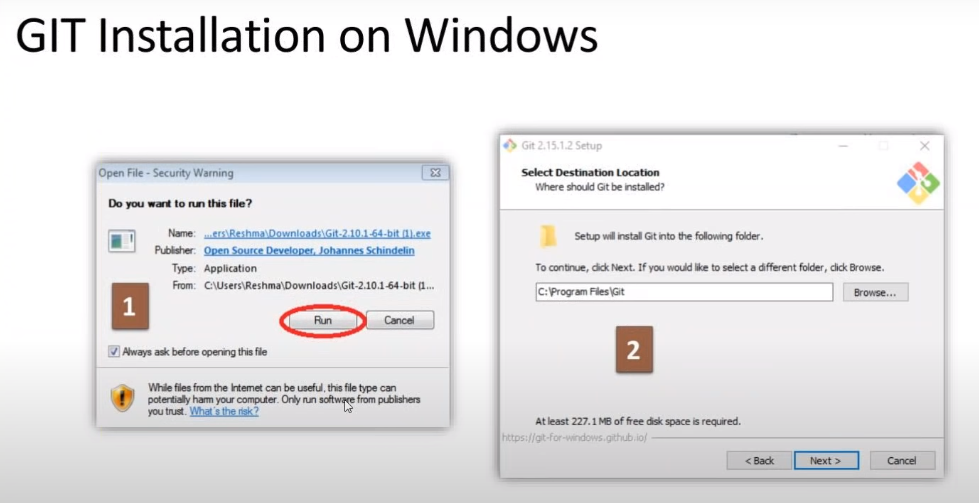
1. Create a repository in github. Doing so, we will get some space in github and also we will be given a url. We can access our project using this url.
2. New Repository > Enter Repository name > Public > Create Repository
3. Note down the url.

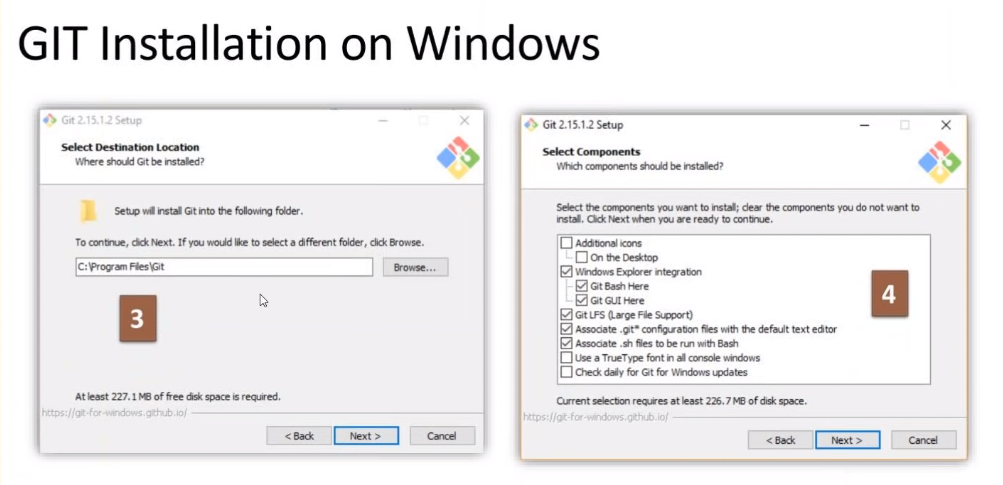


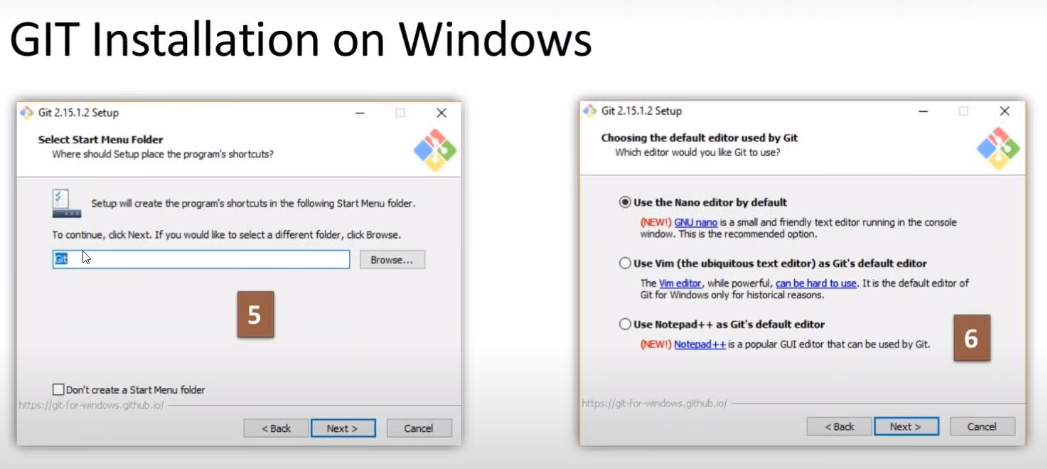
* Installation of git in local, screenshots:

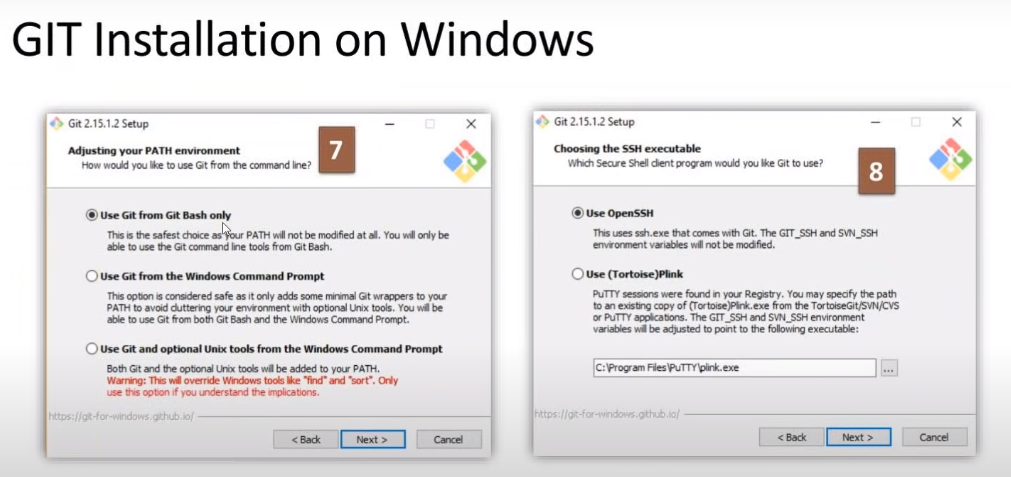


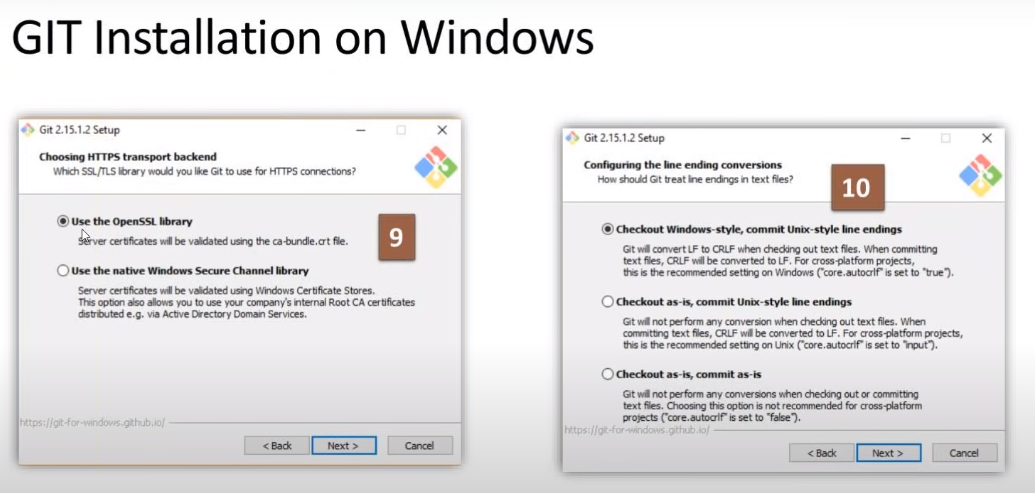


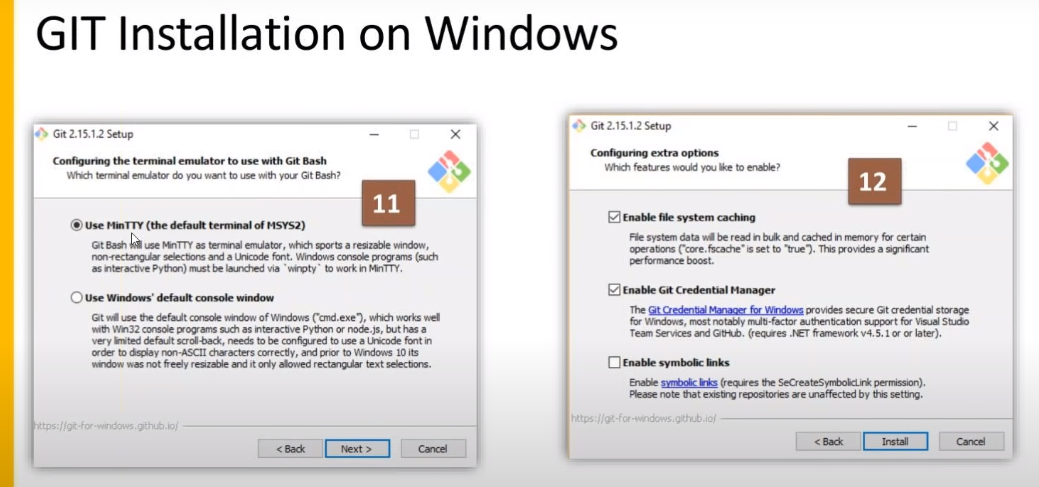


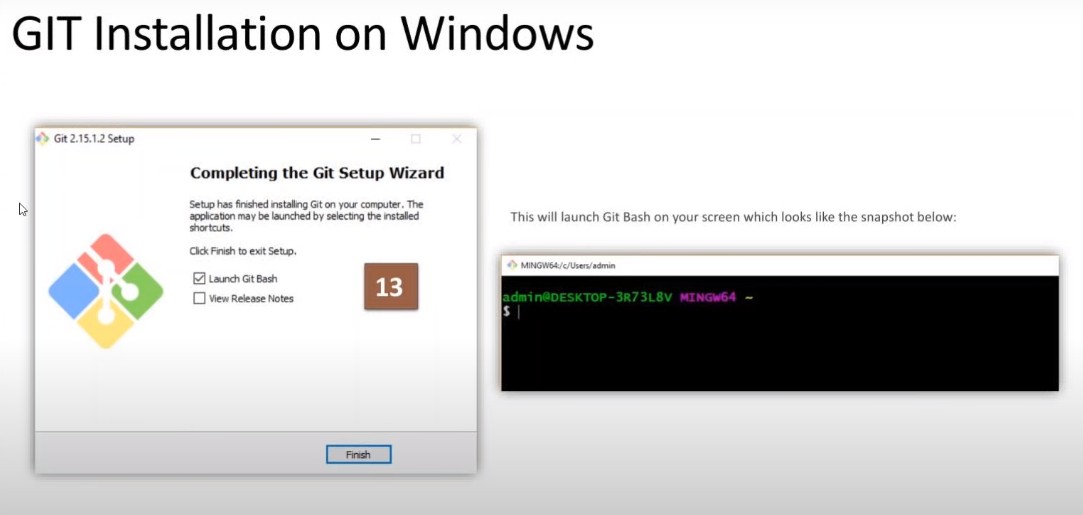




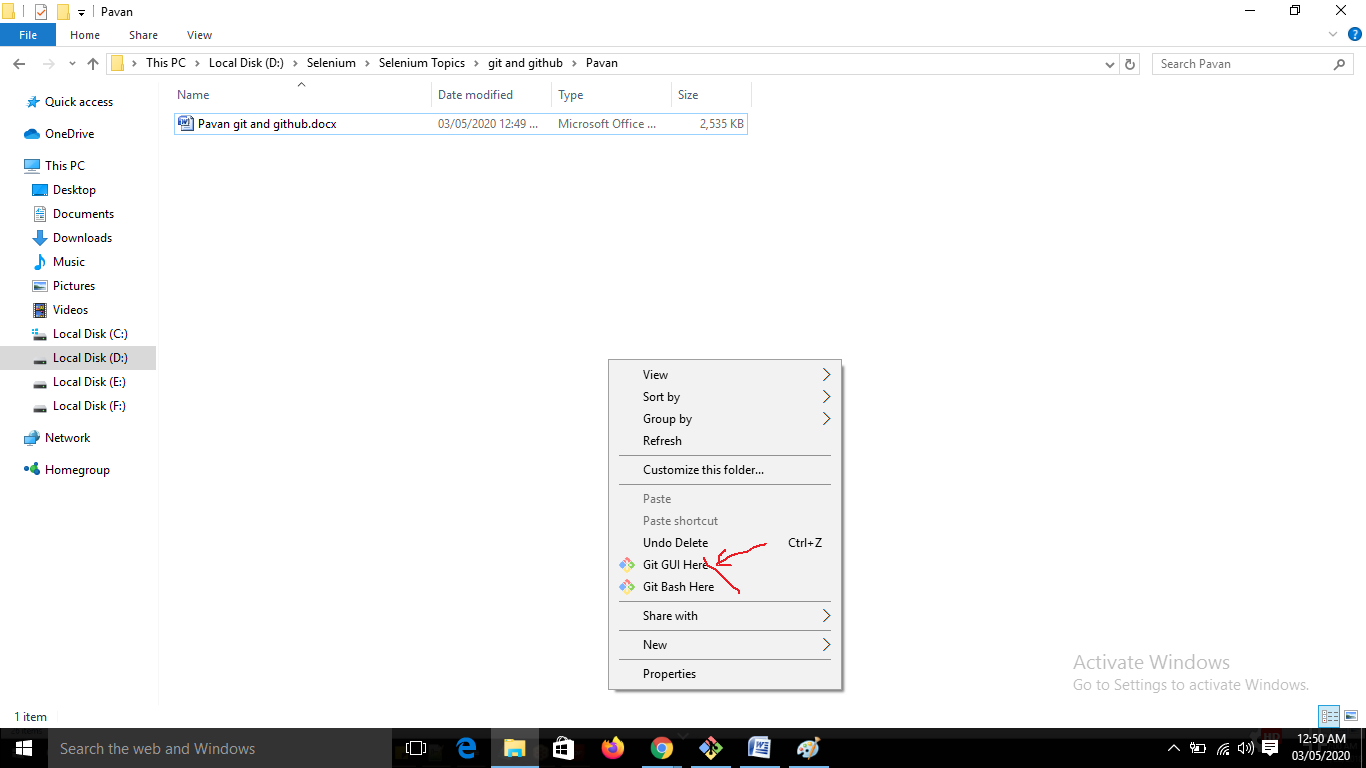




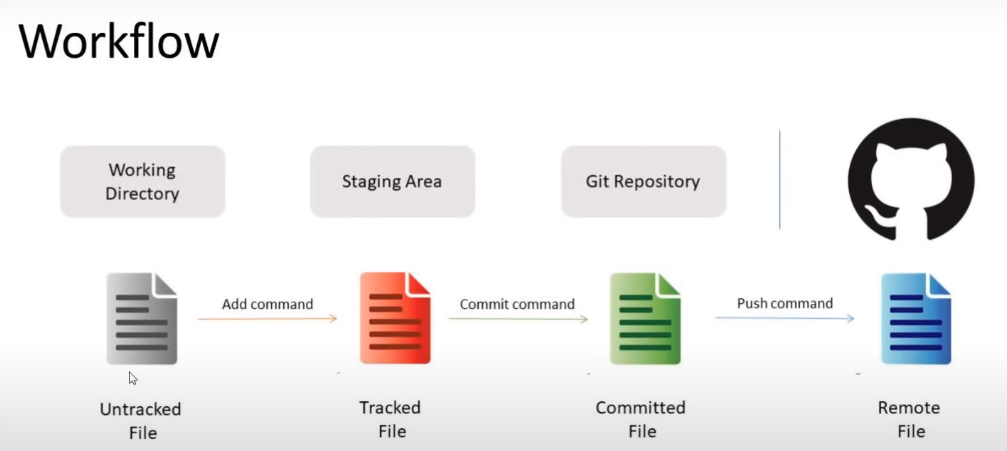




* To make sure git is installed, go to any directory and right click. Check ‘Git GUI Here’ and ‘Git Bash Here’ options are available or not. If available, that means git is installed perfectly.



* Workflow is shown below:



* Git commands:
* Go to project folder.
* Open Git Bash in the project folder.
* Create the local repository.
* git init: Create an empty git repository(Local Repository). .git is the name of the repository.
* Now we have to connect local repository with global.
* git remote add origin “<global repository url>”