**Install git, if not installed already.**

Download link: <https://git-scm.com/downloads>

**Configure git on desktop, if not done already.**

1. Open command prompt.
2. Use the following commands:

*git config --global user.name "[name]"*

*git config --global user.email "[email address]"*

(Use your name instead of “[name]” and the email address used to signup for github instead of “[email address]”)

**Working with the project**

1. Create a new folder called “Career-Information-and-Recruitment-Portal” (without double quotes) at any preferred location in your pc.
2. cd to this folder OR

shift + right click and open PowerShell window (or command prompt) from that directory.

1. Initialize this folder as a git repo using the command:

*git init*

1. **Adding remote origin:**

Use the following command to connect to the online git repo and sync it to your pc.

*git remote add origin [url]*

Instead of [url], use the url for our project repo: <https://github.com/jincy-p-janardhanan/Career-Information-and-Recruitment-Portal.git>

NB: Steps 1 to 4 has to be done only once throughout working with the project.

1. To synchronize your files and folders in the master branch with the online git repository, use the following command -

*git pull*

1. **Using your own branch for working with the project:**

* Create and switch working to a new branch using this command:

*git checkout -b [branch-name]*

Instead of branch-name, provide any descriptive name for your branch.

(This command creates a new branch if it does not exist and switches to it immediately. Alternatively, you can use the following commands to create and switch to the branch.

*git branch [branch-name]*

*git checkout [branch-name]*

Note: the checkout command can be used to switch to any existing branch also.)

* Check on which branch you are currently working using the command:

*git status*

This command also helps to find files that have changes which are not committed.

(Always do this before working with your directory and committing files.)

1. Work on the project. While working, use the following commands to stage files and commit.

* **Staging files:**
  + To stage all files (means all files will be updated and made ready to commit to the current branch.) –

*git add .*

* + To stage only specific files (only the specified file will be updated and made ready to commit to the current branch) –

*git add [filename]*

* **Committing changes:**

*git commit -m “descriptive message about the changes”*

Inside the double quotes briefly mention the changes made to the files.

(The commit command helps to save the current state of the committed file. To make the commit available online on the branch, you have to push it.)

* Checking log (to view all commits to the current branch) –

*git log --oneline*

(This command lists all commits to the current branch with their names and message.)

Or, you can use this command -

*git log*

(This gives a very detailed description of all commits to the current branch.)

* **Pushing changes:**

*git push -u origin [branch-name]*

Use this command to push changes to the current branch.

(The push command uploads all changes to the current branch.)

Note: If you get the following error while pushing -

error: failed to push some refs to 'https://github.com/jincy-p-janardhanan/Career-Information-and-Recruitment-Portal.git'

hint: Updates were rejected because the remote contains work that you do

hint: not have locally. This is usually caused by another repository pushing

hint: to the same ref. You may want to first integrate the remote changes

hint: (e.g., 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

Try using git pull command.

* **After making all changes required** (you may commit many times during the process), **create a pull request** by logging into <github.com>.

(Detailed steps are available [here](https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/creating-a-pull-request).)

Please don’t use git pull from command line because it merges all changes to the master branch, before anyone can review it.

**OTHER USEFUL COMMANDS**

* To view all branches-

*git branch --all*

* To undo a specific commit -

*git revert [commit]*

In place of [commit], use the id of the commit which you have to undo. Commit id will be something like “2a98055”, it is the number generated when you do a commit on any branch. You can see it in the output after the commit command. You can also use the command -

*git log --oneline*

to find commit ids.

* To undo all commits after a specific commit –

*git reset [commit]*

This command undoes all commits after the specified commit.

**QUICK REFERENCES**

* [GitHub Cheat Sheet](https://github.github.com/training-kit/downloads/github-git-cheat-sheet/)