|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Git & Git Hub | | | | | |
|  |  |  |  |  |
|  | | |  | |

**What is GitHub generally, and what is it mainly used for?**

GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. To understand exactly what GitHub is, you need to know two connected principles:

* Version control
* Git

**What Is Version Control?**

Version control helps developers track and manage changes to a software project’s code. As a software project grows, version control becomes essential.

**What Is Git?**

Git is a **specific open-source version control system** created by **Linus Torvalds** in 2005.

Specifically, Git is a **distributed version control system**, which means that the entire codebase and history are available on every developer’s computer, which allows for easy branching and merging.

**So What Is GitHub, Then?**

GitHub is a cloud-based Git repository hosting service. It makes it a lot easier for individuals and teams to use Git for version control and collaboration.

GitHub’s interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technical savvy and use of the command line.

GitHub is so user-friendly, though, that some people even use GitHub to manage other types of projects.

Additionally, anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

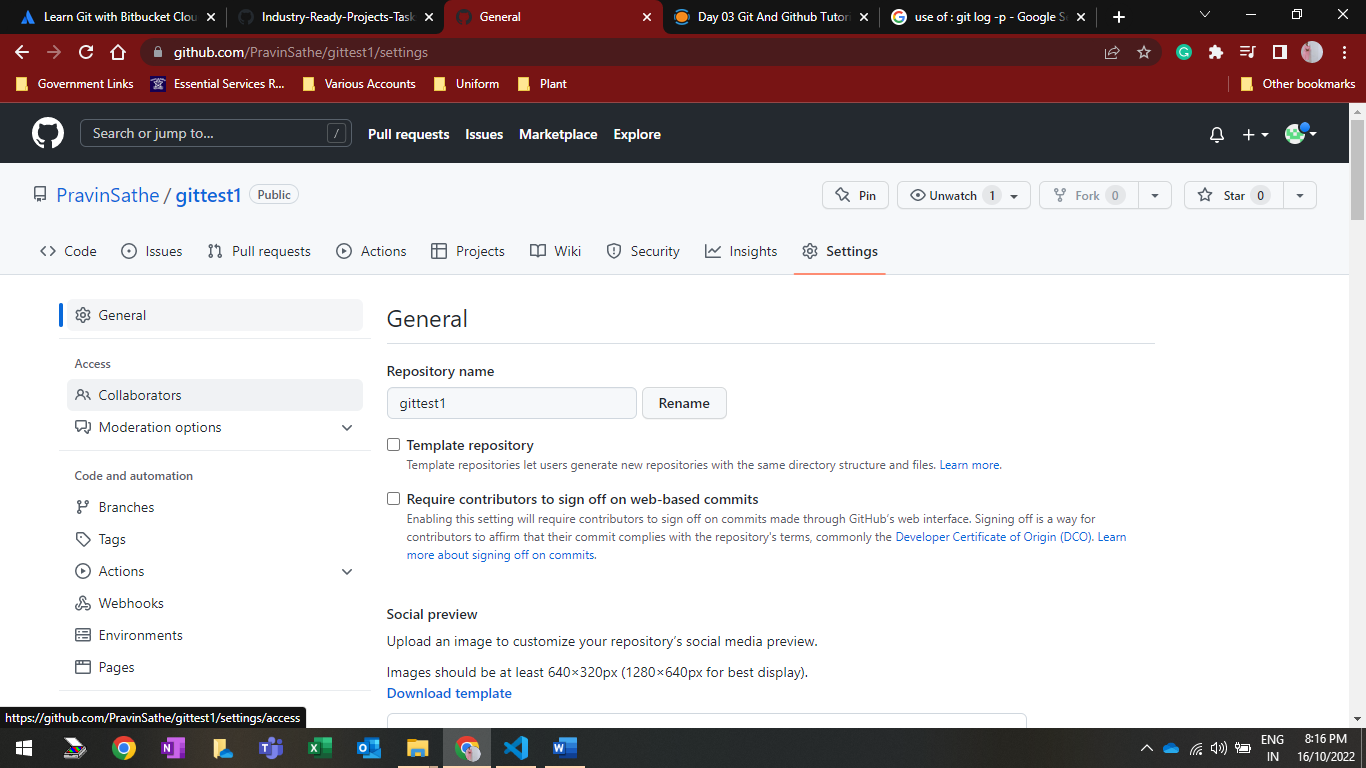
**Task-1 : Git Commands with Example**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Command | Use | Example |
| 1 | git version |  |  |
| 2 | Git status | This command lists all the files that have to be committed |  |
| 3 | git config --global user.name “[name]” | This command sets the author name to be used with your commits. |  |
| 4 | git config --global user.email “[email address]” | This command sets the email address respectively to be used with your commits. |  |
| 5 | git init [repository name] | This command is used to start a new repository. |  |
| 6 | git add [file] | This command adds a file to the staging area. |  |
| 7 | git add . | This command adds one or more to the staging area |  |
| 8 | git help -a | Take help from github help section for different commands and other errors. |  |
| 9 | mkdir store | Create a directory if not created initially. |  |
| 10 | cd store | To go inside the directory and work upon its contents. |  |
| 11 | git commit -m "first commit" | To commit our changes(taking a snapshot) and providing a message to remember for future reference. |  |
| 12 | git branch | To see all the branches present and current branch that we are working on. |  |
| 13 | git branch -M main |  |  |
| 14 | git remote add origin https://github.com/PravinSathe/gittest1.git |  |  |
| 15 | git remote -v |  |  |
| 16 | git push -u origin main |  |  |
| 17 | git restore --staged . |  |  |
| 18 | git branch [developer1] |  |  |
| 19 | git checkout [developer1] |  |  |
| 20 | git merge [developer1] |  |  |
| 21 | git logs |  |  |
| 22 | git log -p | Git log helps you see the past commits which helps to see who did what in Git and the repository. | [Screenshot not available] |
| 23 | git branch [-d developer1] |  |  |
| 24 | git clone [https://github.com/PravinSathe2/gittest1.git] |  |  |

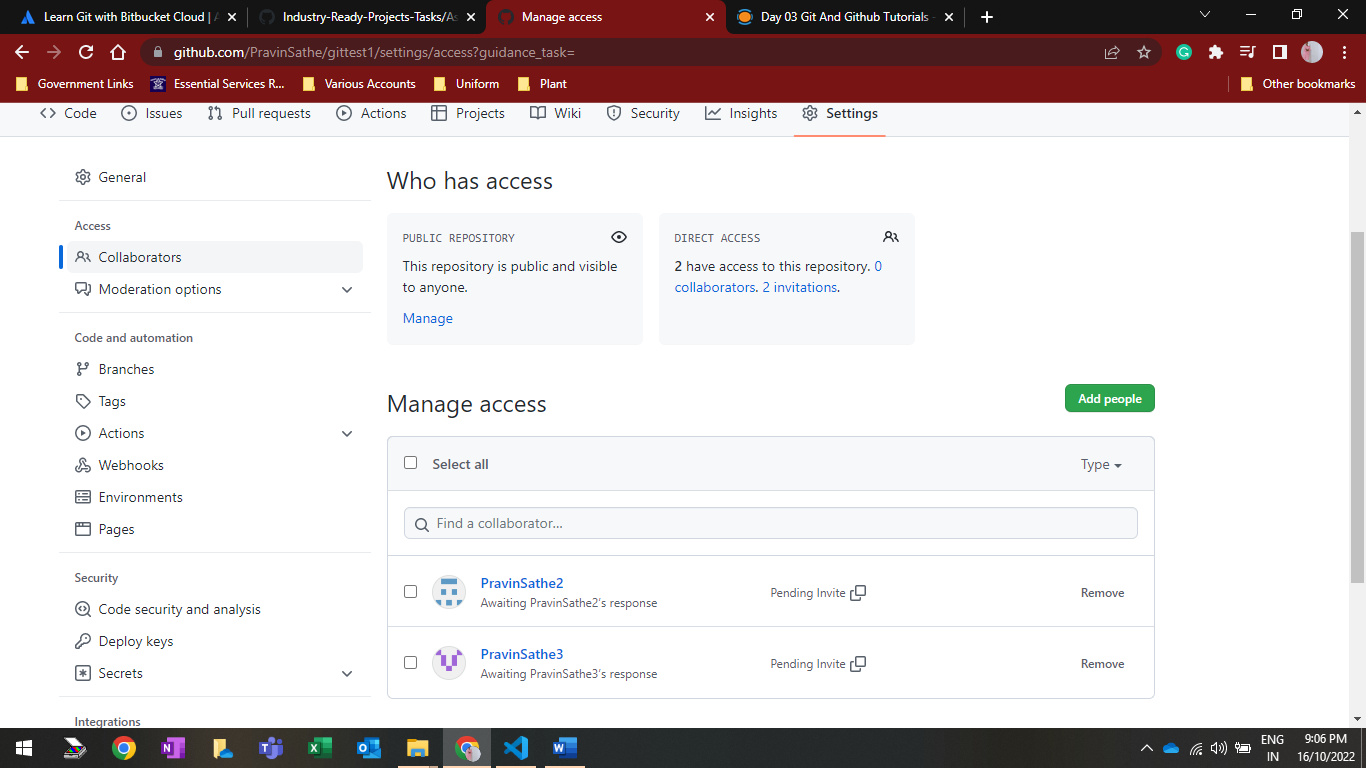
**Task-2: Git Commands with Example**

Consider that your want to start an open-source project in your organization. Perform all the standard operation to create a repository with minimal permission for all the users. It should contain.

* Proper open source structure
* Proper Readme
* Add 2 collaborator
* Host GitHub Pages using settings (Designed to host your personal, organization, or project pages from a GitHub repository)
* **Add Collaborator:**



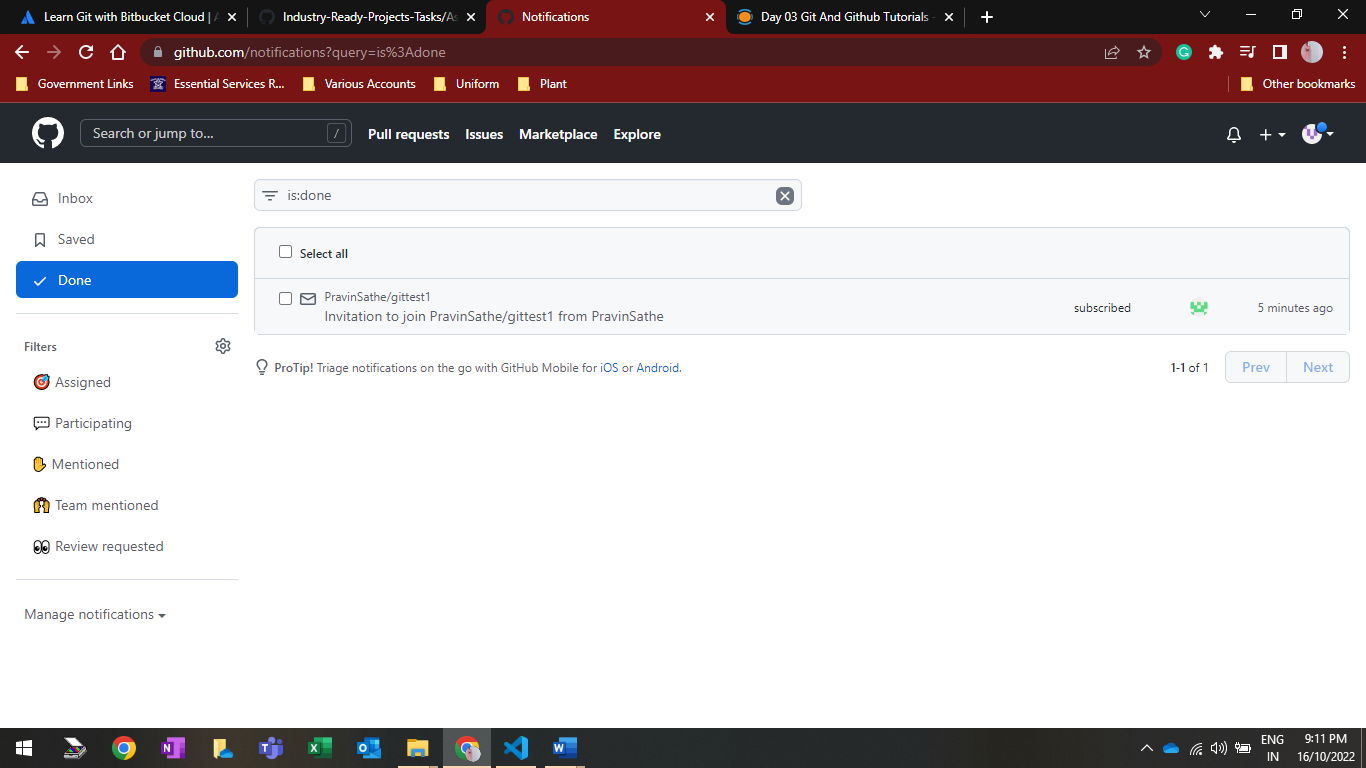




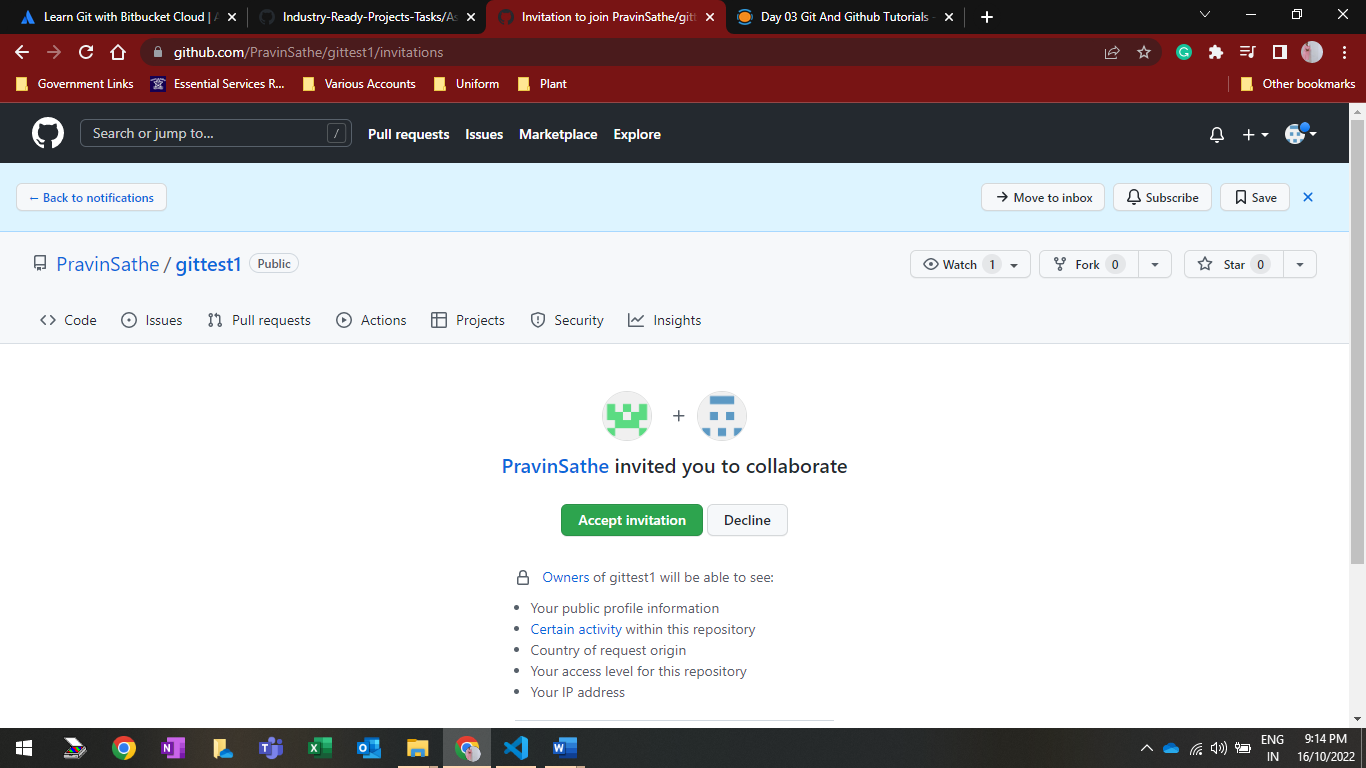


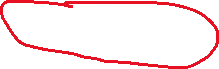
* **Accepting Collaborate request:**

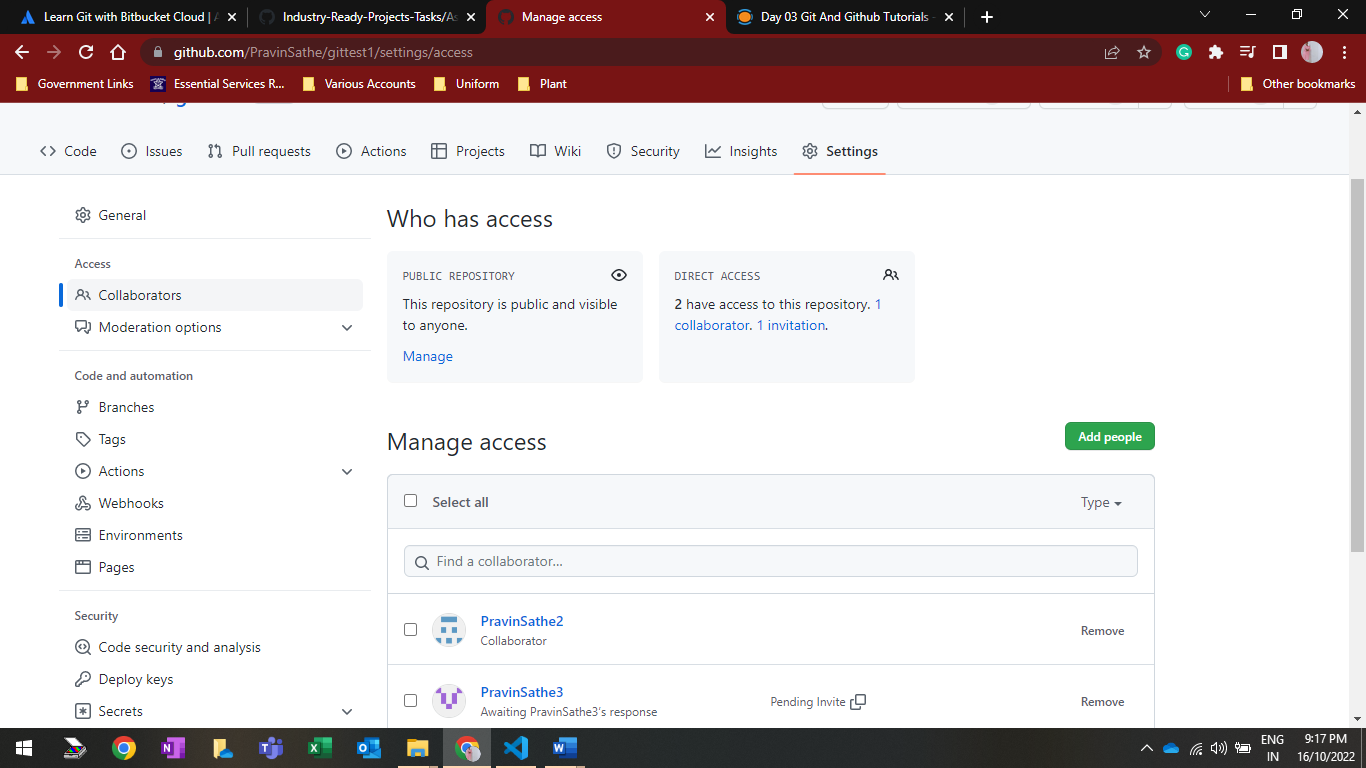














* **Fork repository for access the content**



