Subject Name:

**Source Code Management**

Subject Code:

**CS181**

Cluster:

**BETA**

Department:

**DCSE**



**Submitted By:**

**Submitted To :-**



Abhinav Rawat Mr. Monit Kapoor

2110990055

G01

Team Members:

Abhimanyu Kashyap - 2110990042

Aaryan Prabhakar - 2110990018



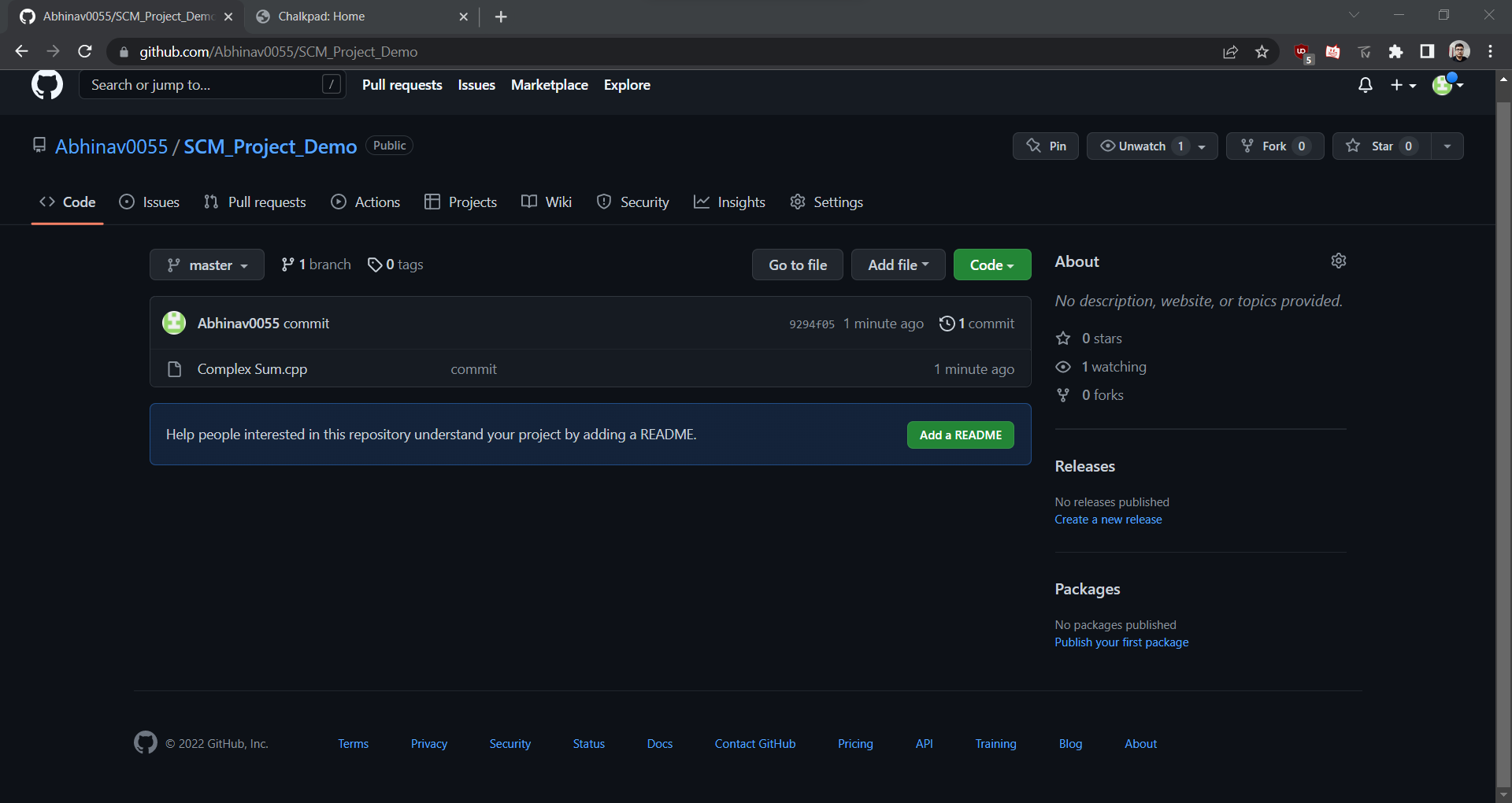
**INDEX**

|  |  |
| --- | --- |
| **S. No.** | **Task Title** |
| 1. | Add collaborators on GitHub Repository |
| 2. | Fork and Commit |
| 3. | Merge and Resolve conflicts |
| 4. | Reset and revert |

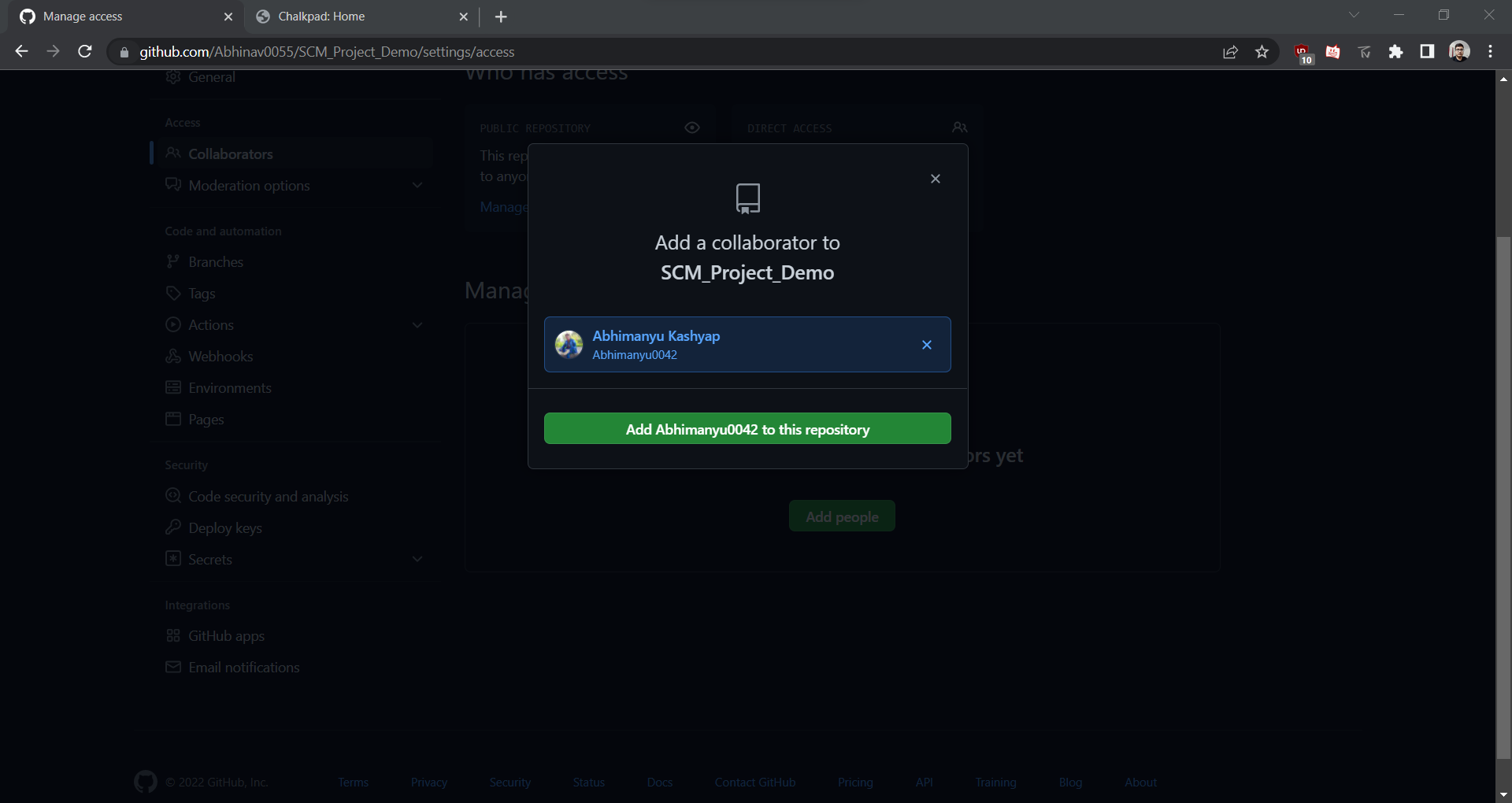
**Experiment No. 01**

# **Add collaborators on GitHub Repository: -**

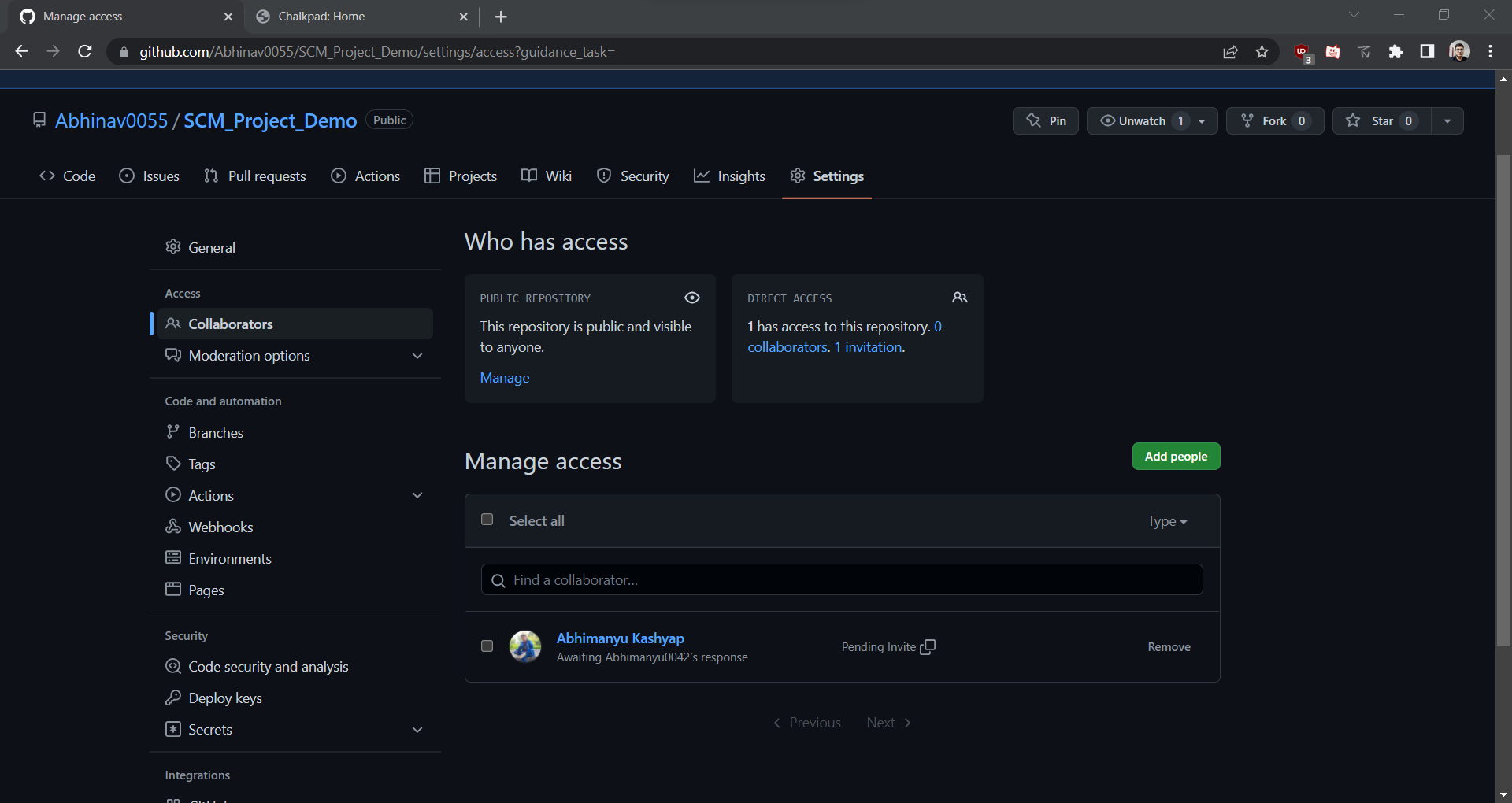
1. Add a collaborator on the existing GitHub Repository.

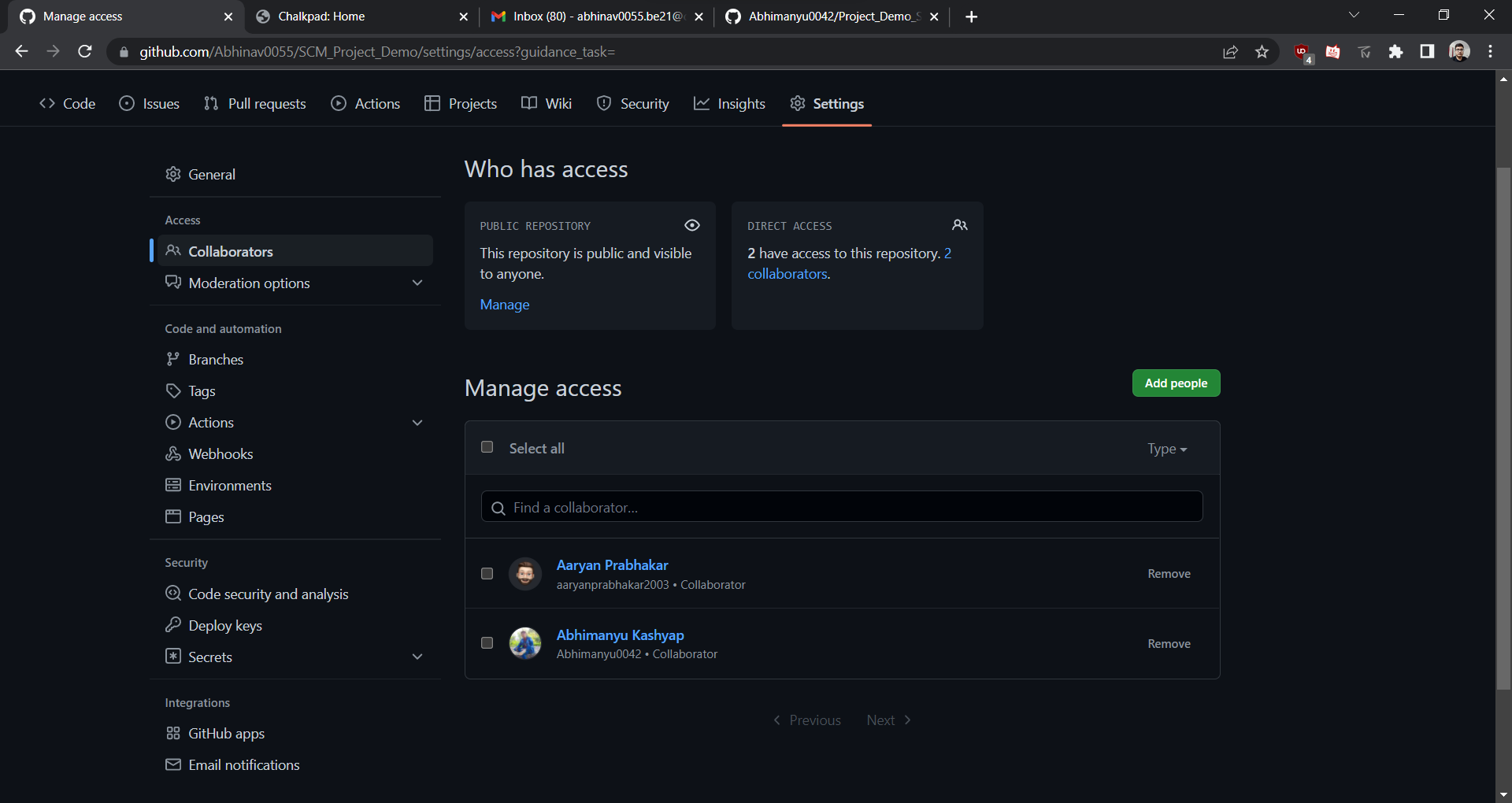


1. Go to Collaborators in Repository Setting, add the username or email of Collaborator you want to add in your Repository.



1. Invitation Mail is sent to the Collaborator. After the collaborator has to accept this Invitation.





1. New Collaborator can now access the repository.

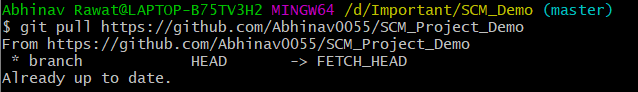
**Experiment No. 02**

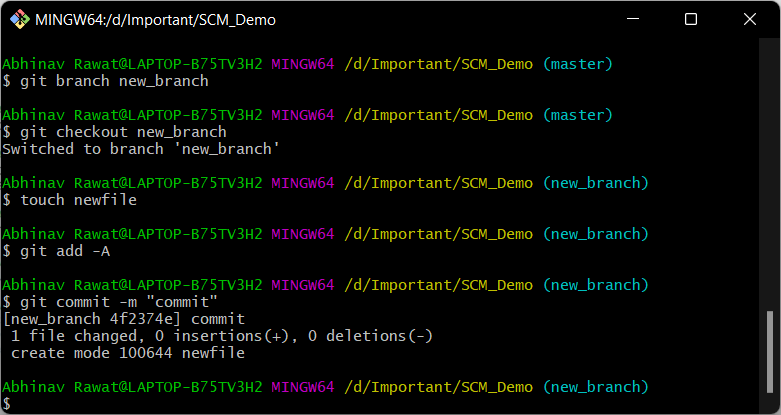
**Fork and Commit:-**



1. Create a New Repository (or Folder) and Initialized the Git repository, Using ‘git init’ command. And add origin of your remote Repository to local Repository.

Git pull <URL> This command is used to fetch the remote repository or to clone the repository.

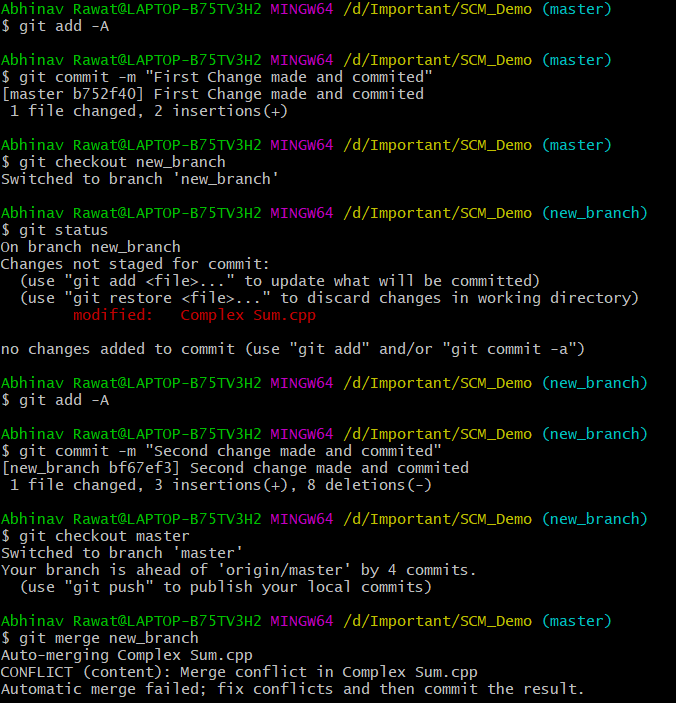
1. Type “Git pull <link of repository>” on CLI.

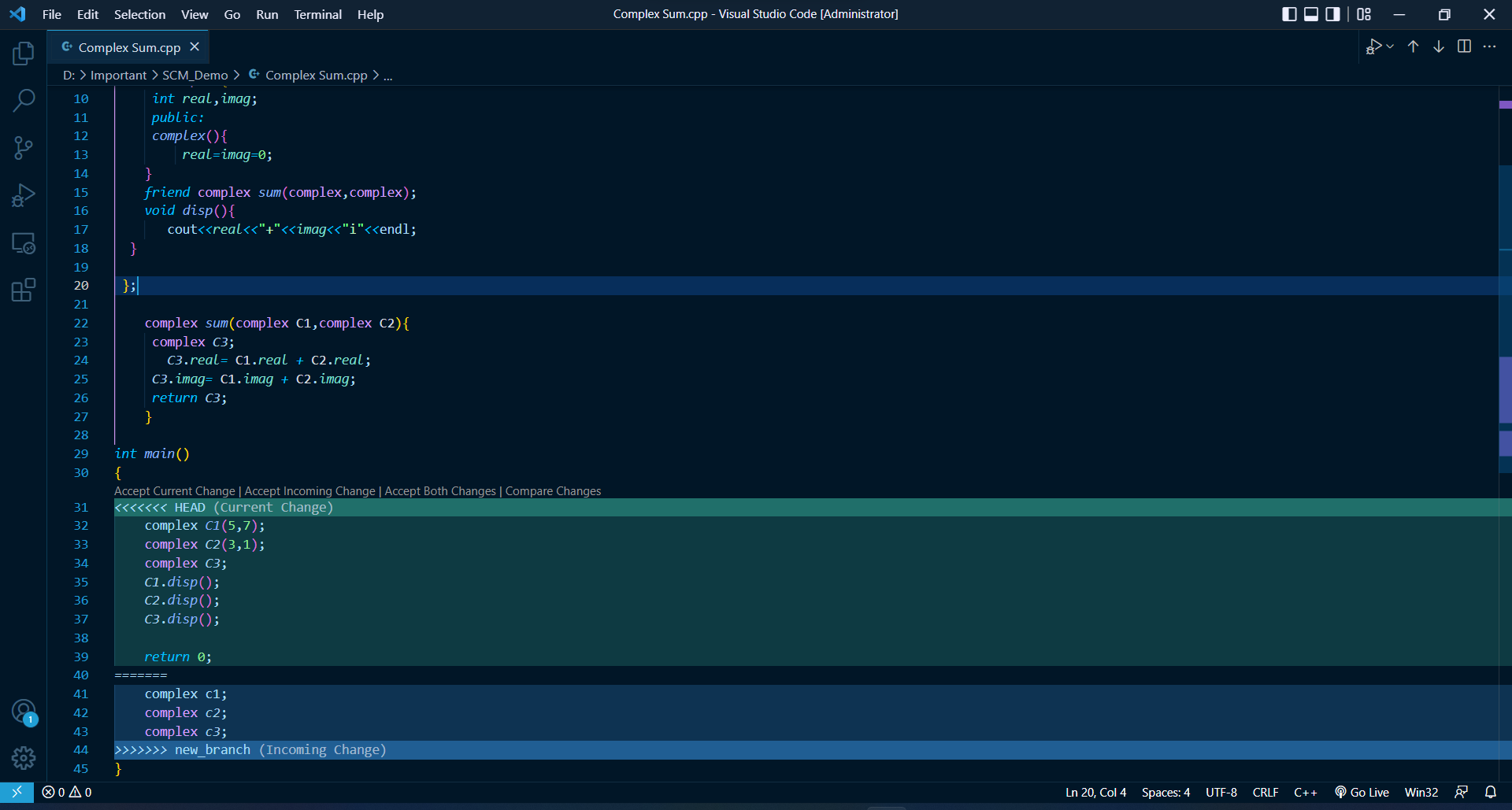
1. Create a new Branch <name>. Create a new file and do changes in it and commit it.
2. Commit is made on the fork Repository

**Experiment No. 03**

**Merge and Resolve conflicts created due to own activity and collaborators activity :-**

1. Do changes in master branch and commit those change. And checkout to the new branch and again do changes and commit it. Now checkout to master branch and merge the branch in master.

Now if any conflict arises due to the activities performed, open the merge tool using $git mergetool now resolve the conflicts.

Press “I” to insert. After insertion, Press “: wq”. The merge conflict is solved and our branch is merged to master branch.

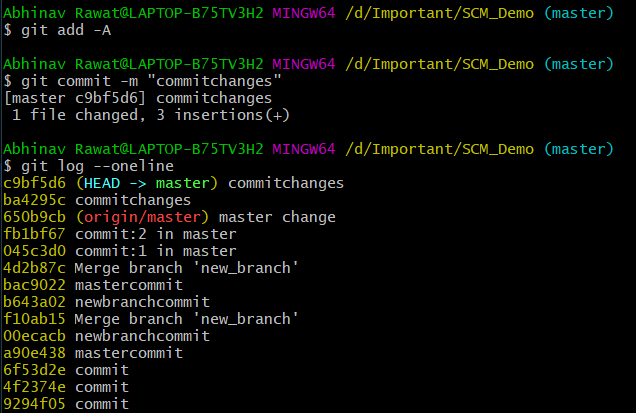
**Experiment No. 04**

Reset and Revert

## Git Revert

The git revert command is essentially a reverse git cherry-pick. It creates a new commit that applies the exact opposite of the change introduced in the commit you’re targeting, essentially undoing or reverting it

1. Do some changes to a file and Commit them.



On Git Bash CLI, Type command “git Command <Commit id>”. It reverts the changes that done before Commit.

## **Git Reset :-**

RESET is the command we use when we want to move the repository back to a previous commit, discarding any changes made after that commit.

1. Type command “git reset –soft <Commit ID>”.

