A Project report on

“Acting Theatre - LCA”

with

Source Code Management

(CS181)

Submitted by

Team Member 1 Name :- **SIDDHARTH SINGH** Roll No. **2110991360**

Team Member 2 Name :- **SHUBHAM** Roll No. **2110991337**

Team Member 3 Name :- **YASH KAPIL** Roll No. **2110991875**

Team Member 4 Name :- **SIDHYANT** Roll No. **2110991363**

Department of Computer Science & Engineering

Chitkara University Institute of Engineering and Technology, Punjab

Jan- June
(2021-22)

Institute/School Name	**Chitkara University Institute of Engineering and Technology**		
Department Name	**Department of Computer Science & Engineering**		
Programme Name	**Bachelor of Engineering (B.E.), Computer Science & Engineering**		
Course Name	**Source Code Management**	Session	**2021-22**
Course Code	**CS181**	Semester/Batch	**2nd/2021**
Vertical Name	**Alpha**	Group No	G24
Course Coordinator	**Dr. Neeraj Singla**		
Faculty Name	**Prof./Dr./Mr./Ms. GAGANDEEP KAUR**		

Submission

Name:

Signature:

Date:

CHITKARA UNIVERSITY

TABLE OF CONTENT

1. Version control with Git

A version control system is a software that tracks changes to a file or set of files over time so that you can recall specific versions later. It also allows you to work together with other programmers.

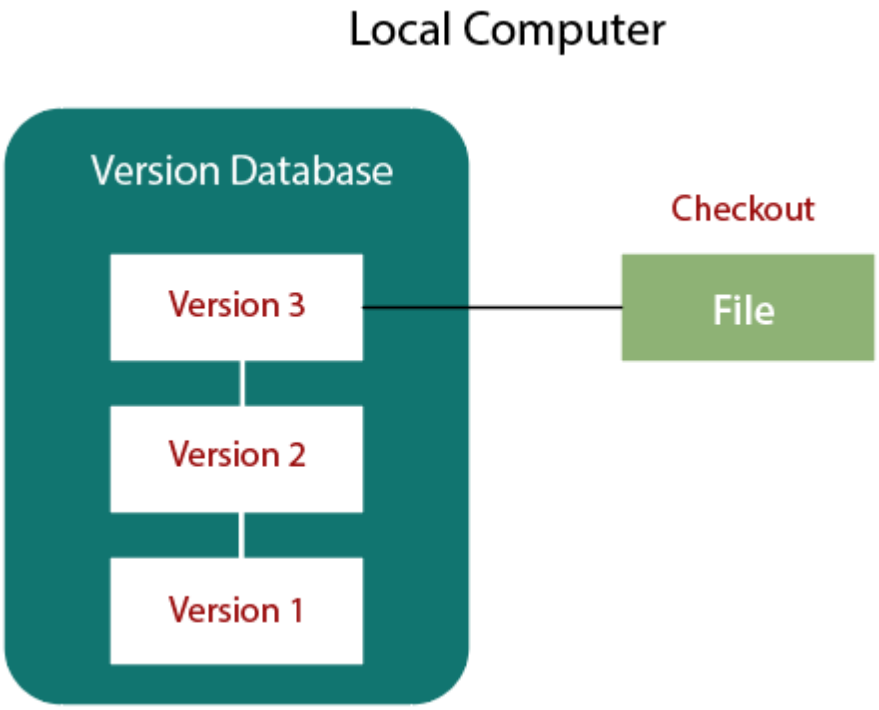
The version control system is a collection of software tools that help a team to manage changes in a source code. It uses a special kind of database to keep track of every modification to the code.

Developers can compare earlier versions of the code with an older version to fix the mistakes.

Types of Version Control System

- Localized version Control System
- Centralized version control systems
- Distributed version control systems

Localized Version Control Systems



The localized version control method is a common approach because of its simplicity. But this approach leads to a higher chance of error. In this approach, you may forget which directory you're in and accidentally write to the wrong file or copy over files you don't want to.

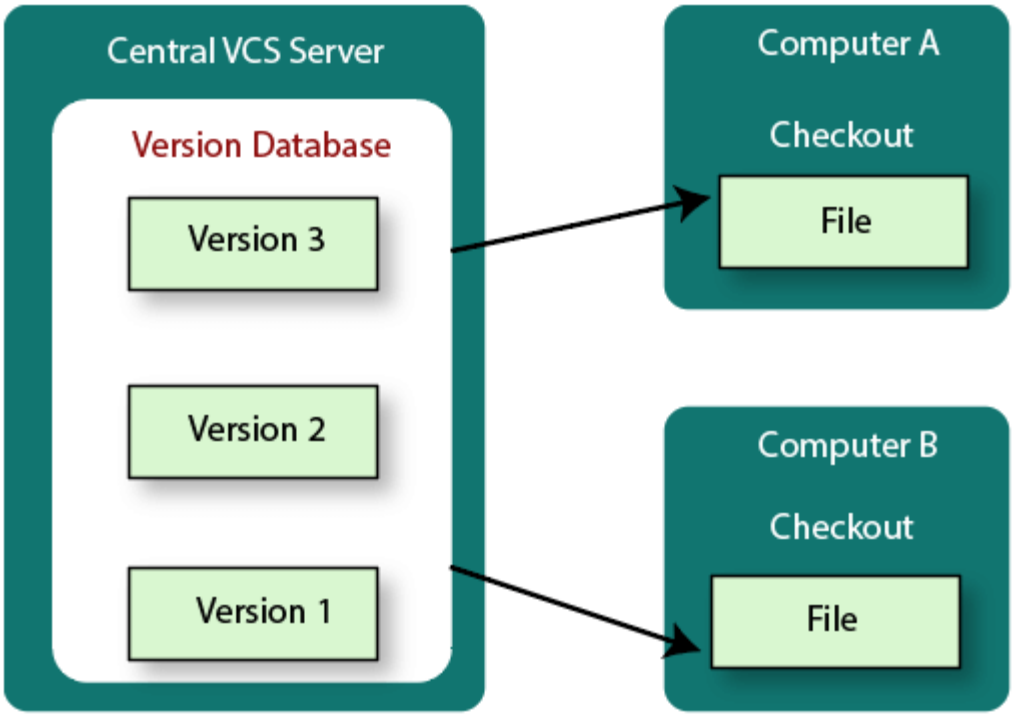
To deal with this issue, programmers developed local VCSs that had a simple database. Such databases kept all the changes to files under revision control. A local version control system keeps local copies of the files.

The major drawback of Local VCS is that it has a single point of failure.

Centralized Version Control System

The developers needed to collaborate with other developers on other systems. The localized version control system failed in this case. To deal with this problem, Centralized Version Control

Systems were developed.



These systems have a single server that contains the versioned files, and some clients to check out files from a central place.

Centralized version control systems have many benefits, especially over local VCSs.

Centralized version control systems have many benefits, especially over local VCSs.

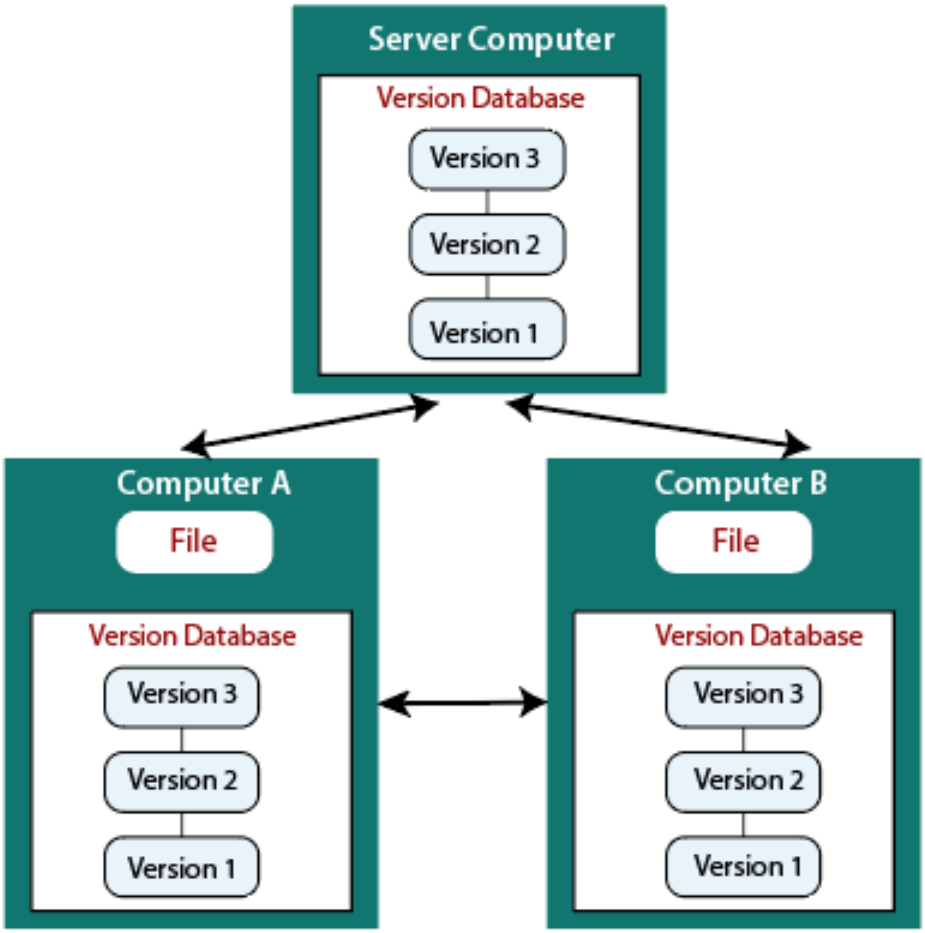
- Everyone on the system has information about the work what others are doing on the project.
- Administrators have control over other developers.
- It is easier to deal with a centralized version control system than a localized version control system.
- A local version control system facilitates with a server software component which stores and manages the different versions of the files.

Distributed Version Control System

Centralized Version Control System uses a central server to store all the database and team collaboration. But due to single point failure, which means the failure of the central server, developers do not prefer it. Next, the Distributed Version Control System is developed.

In a Distributed Version Control System (such as Git, Mercurial, Bazaar or Darcs), the user has a

local copy of a repository. So, the clients don't just check out the latest snapshot of the files even they can fully mirror the repository. The local repository contains all the files and metadata present in the main repository.



DVCS allows automatic management branching and merging. It speeds up of most operations except pushing and pulling. DVCS enhances the ability to work offline and does not rely on a single location for backups. If any server stops and other systems were collaborating via it, then any of the client repositories could be restored by that server. Every checkout is a full backup of all the data.

2.Problem Statement

Create a pull request on a team member's repo and close pull requests generated by team members on own Repo as a maintainer.

3.Objective

Create a distributed Repository and add members in project team and open and close a Pull request.

4.Resource Requirement- Frontend/backend

1.Gitbash

What is Git Bash?

Git Bash is an application for Microsoft Windows environments which provides an emulation layer for a Git command line experience. Bash is an acronym for Bourne Again Shell. A shell is a terminal application used to interface with an operating system through written commands. Bash is a popular default shell on Linux and macOS. Git Bash is a package that installs Bash, some common bash utilities, and Git on a Windows operating system.

Git Bash Commands

Git Bash is packaged with additional commands that can be found in the /usr/bin directory of the Git Bash emulation. Git Bash can actually provide a fairly robust shell experience on Windows. Git Bash comes packaged with the following shell commands which are outside the scope of this document: Ssh, scp, cat, find.
In addition the previously discussed set of Bash commands, Git Bash includes the full set of Git core commands discussed through out this site. Learn more at the corresponding documentation pages for git clone, git commit, git checkout, git push, and more.

2.GitHub

What is GitHub used for?

GitHub allows software developers and engineers to create remote, public-facing repositories on the cloud for free. Once you've set up a repository on GitHub, you can copy it to your device, add and modify files locally, then "push" your changes back to the repository where your changes are displayed to the public.

Key reasons to use GitHub :

Enhanced Collaboration

The single biggest selling point of GitHub is its set of project collaboration features, including version control and access control.

To illustrate what's possible with GitHub, imagine this scenario: You want to code up an online game, and you enlist your friend to help you. You create a **repository** on GitHub that stores all the files, including current and past versions, then give your friend collaborator access to this repo as well.

Easy File Management

GitHub adds a sleek graphical user interface (GUI) layer on top of Git. On its own, Git operates through the command line (a computer's text-based interface). Developers know how to use the command line, but for many, it's not always the most efficient way to interact with files.

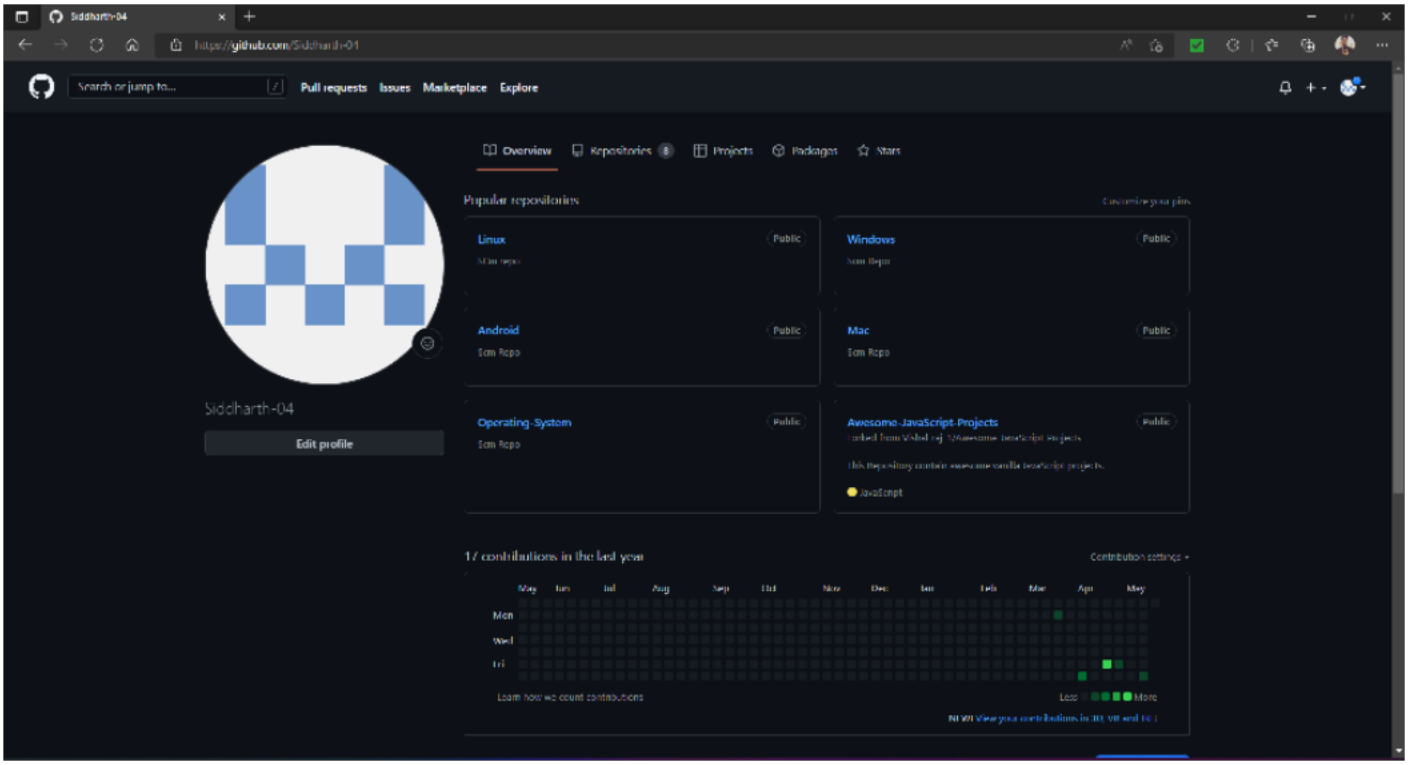
GitHub's interface provides a clean and user-friendly means to perform Git actions as well as view file history. This is more convenient for developers and more accessible for beginners getting the hang of Git.

Social Networking

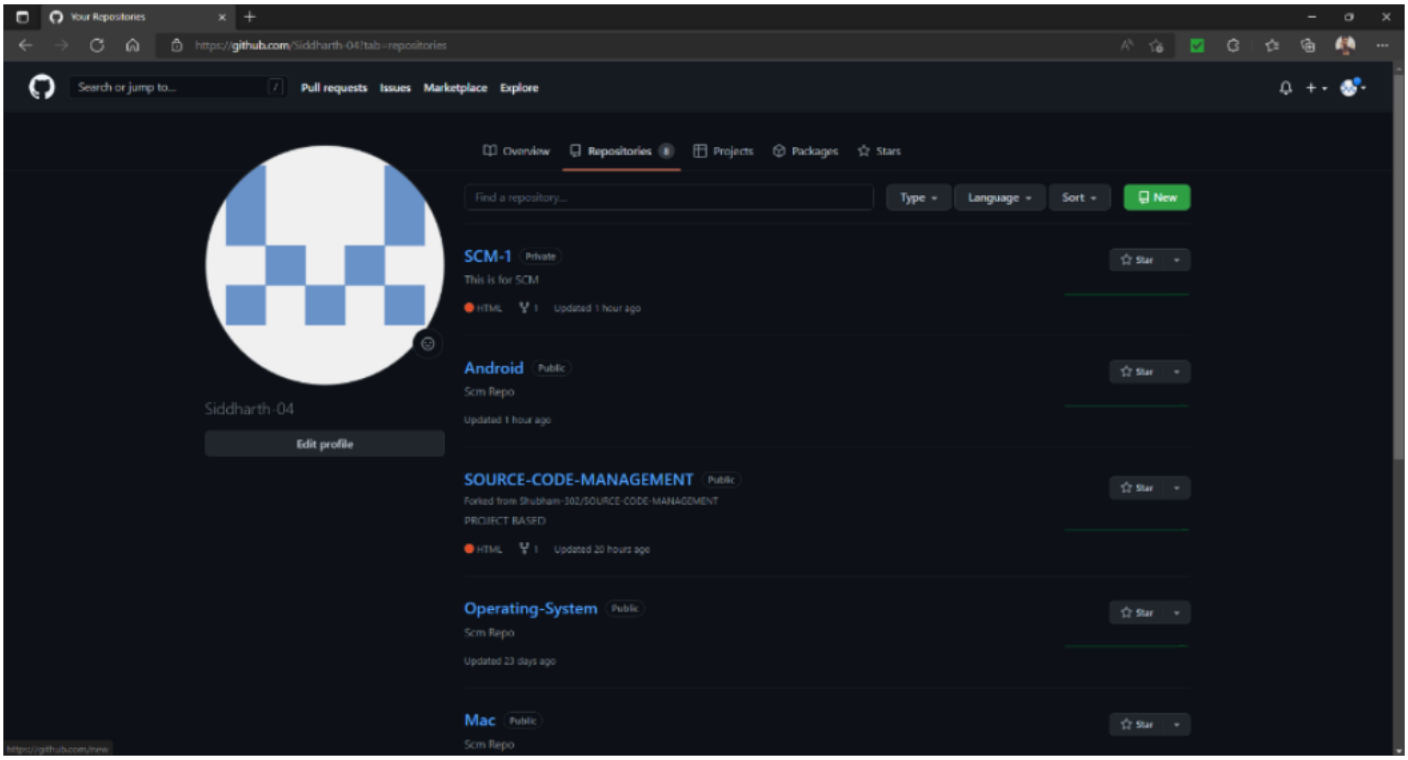
Any GitHub user knows the platform is more than just a place to work on code. All GitHub users have profiles to display their projects, contributions, and activity on the site, and can see anyone's public-facing profile and repositories.

5. Concept and Commands

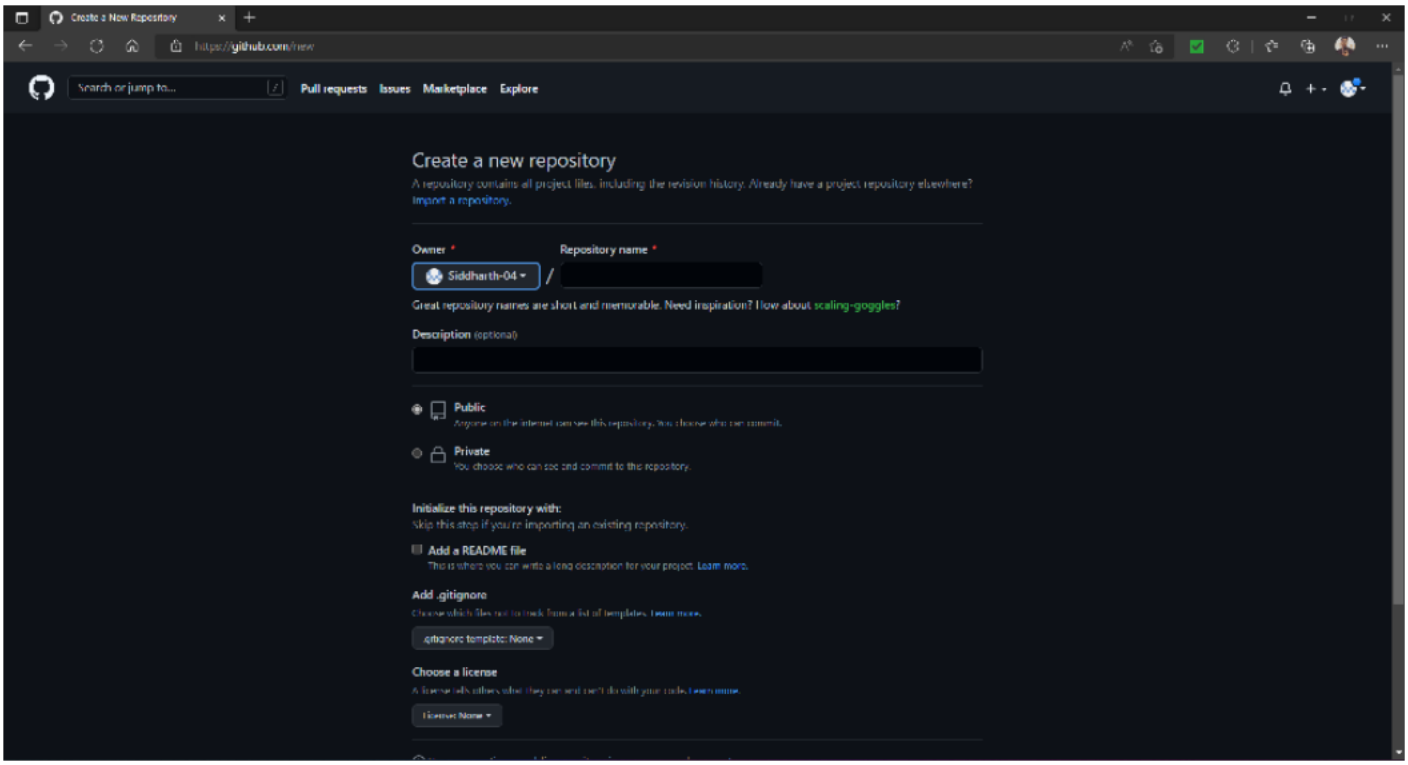
- Login to your GitHub account and you will land on the homepage as shown below. Click on Repositories option in the menu bar.



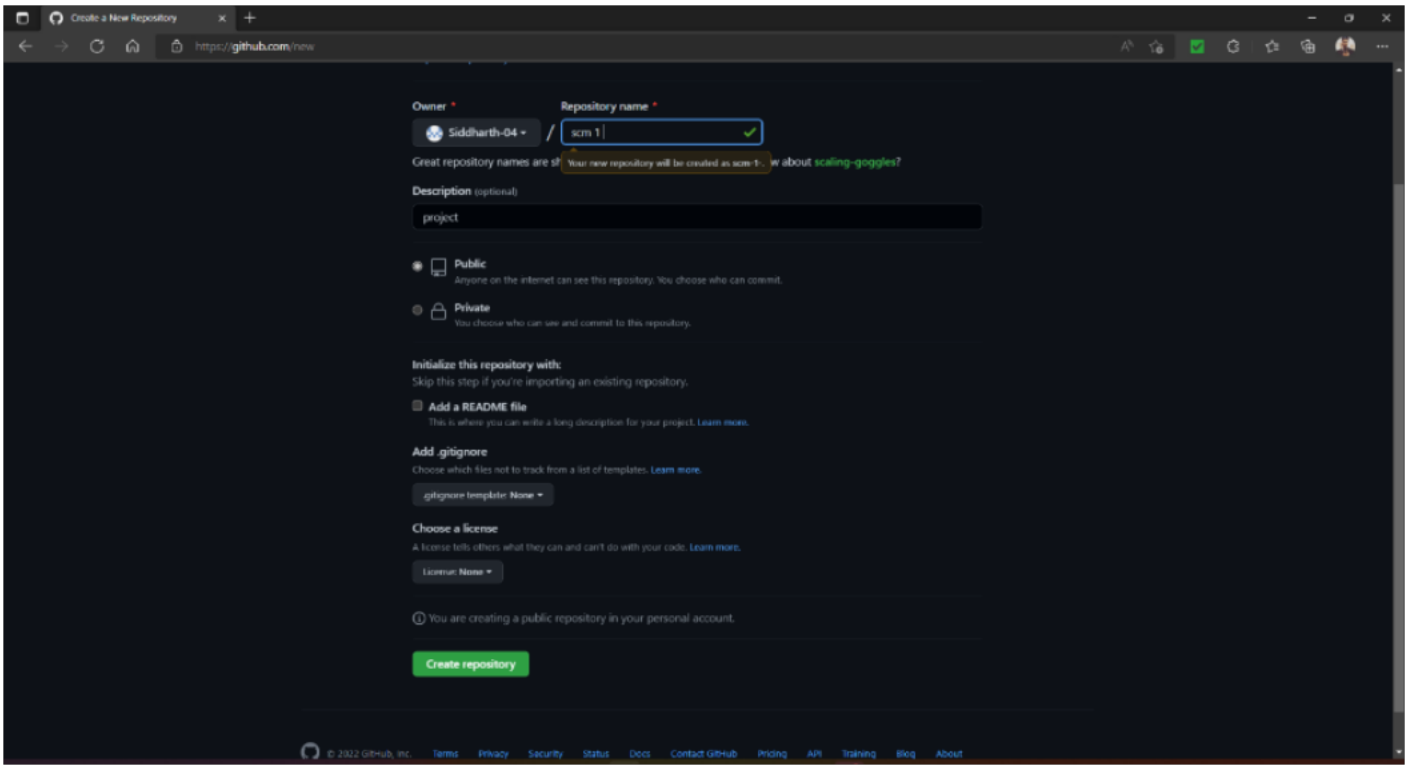
- Click on the ‘New’ button in the top right corner.



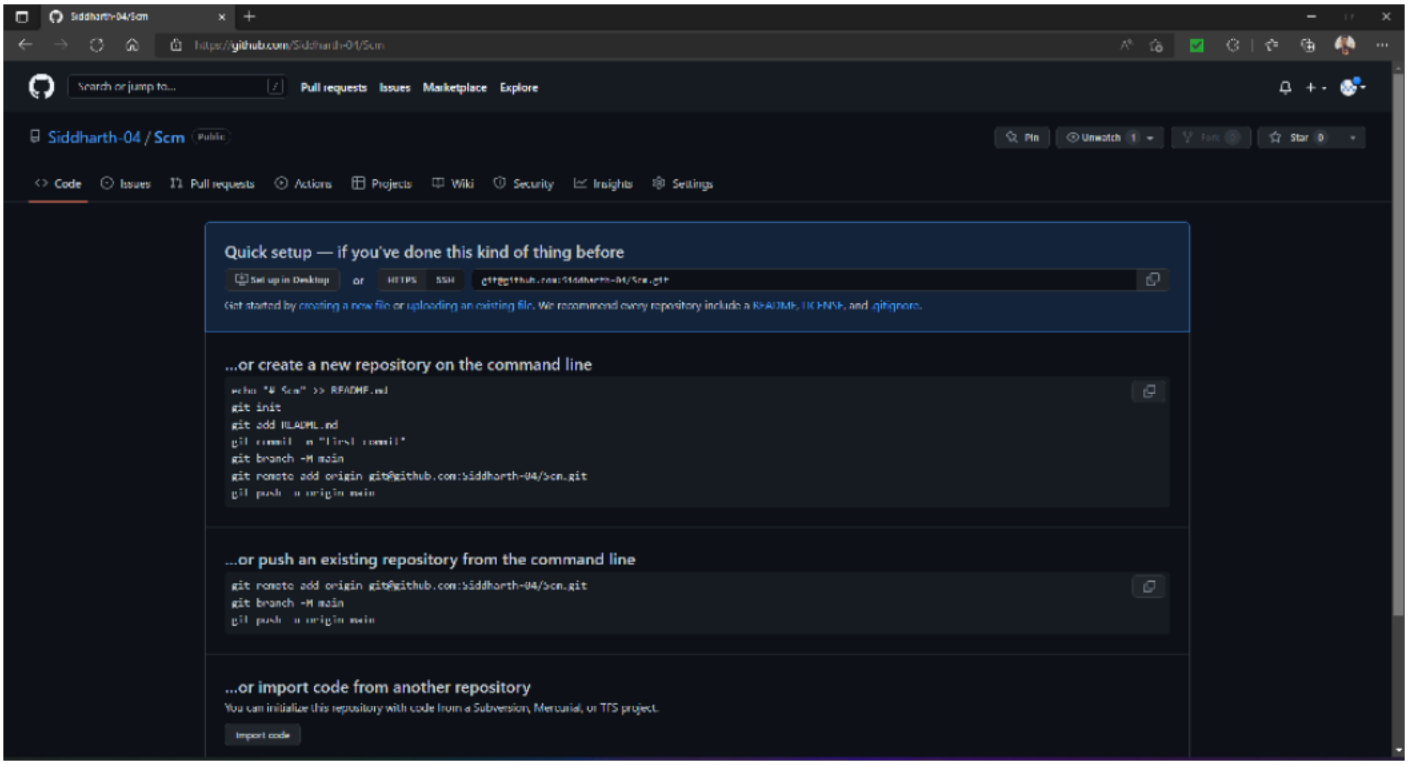
- Enter the Repository name and add the description of the repository.



- Select if you want the repository to be public or private.
- Click to the create repository.

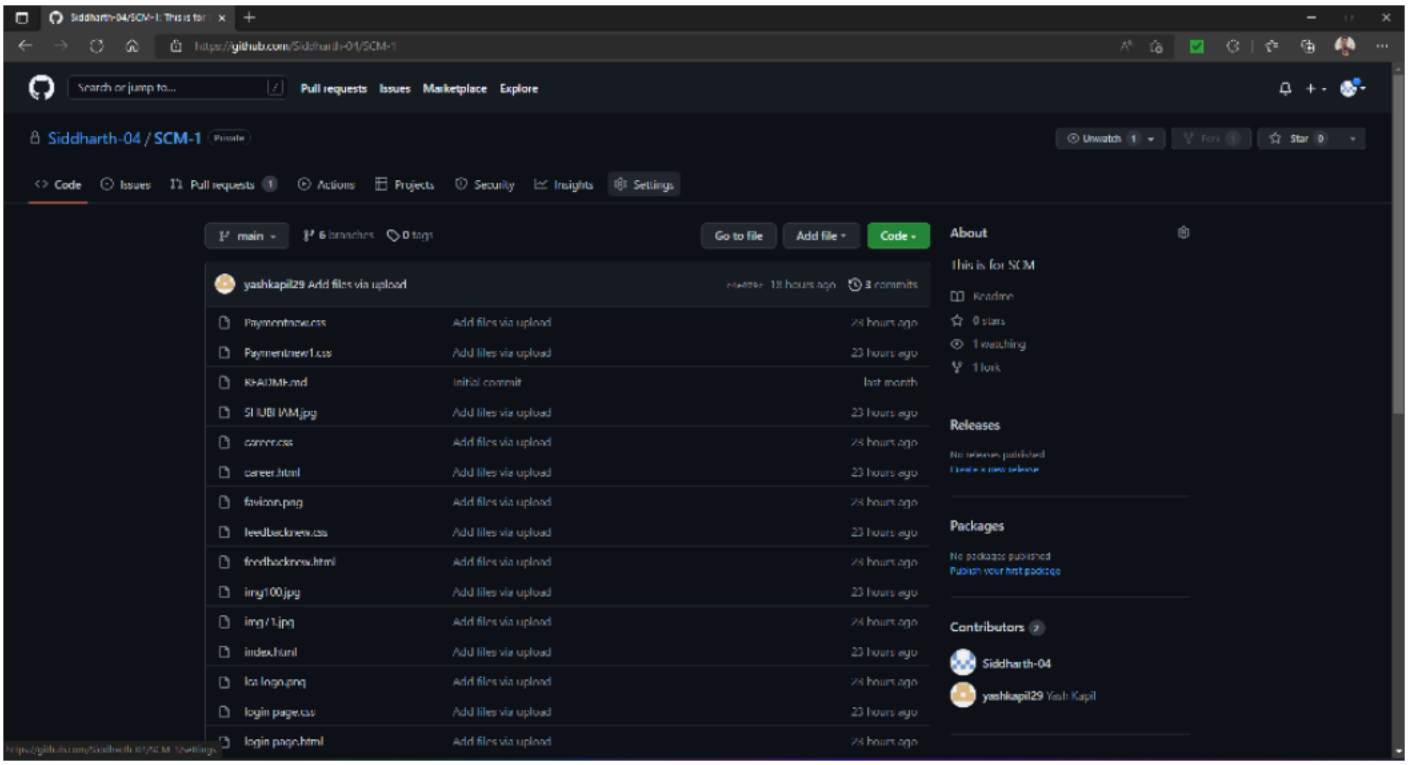


- If you want to import code from an existing repository select the

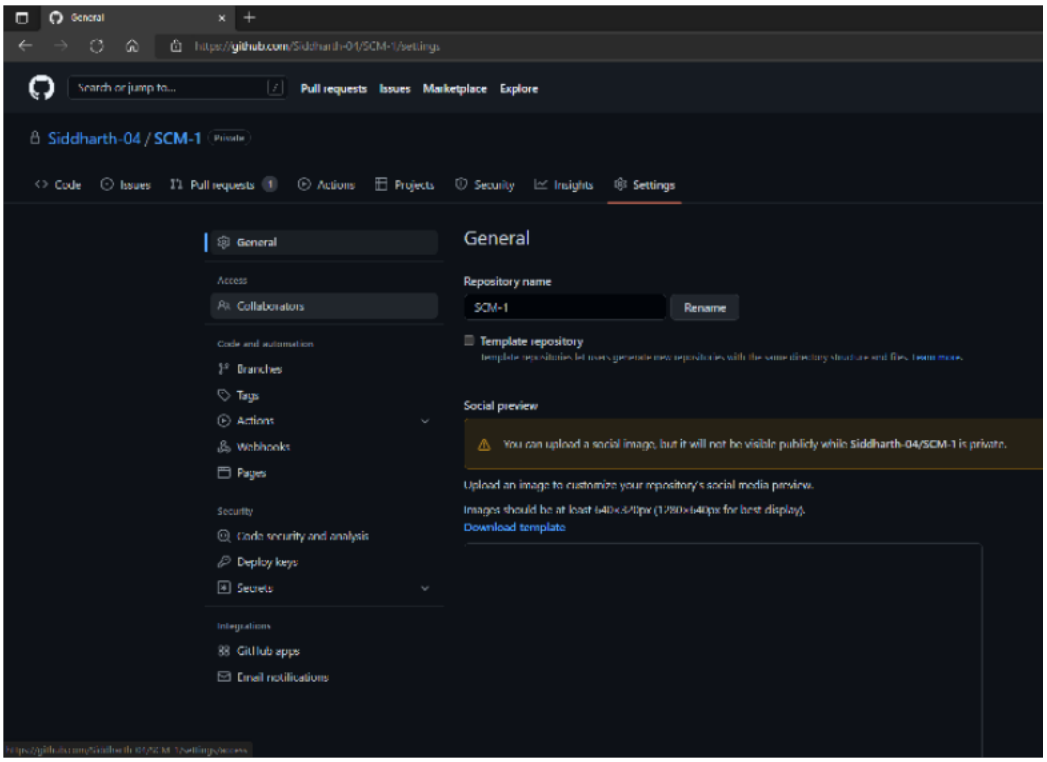


import code option.

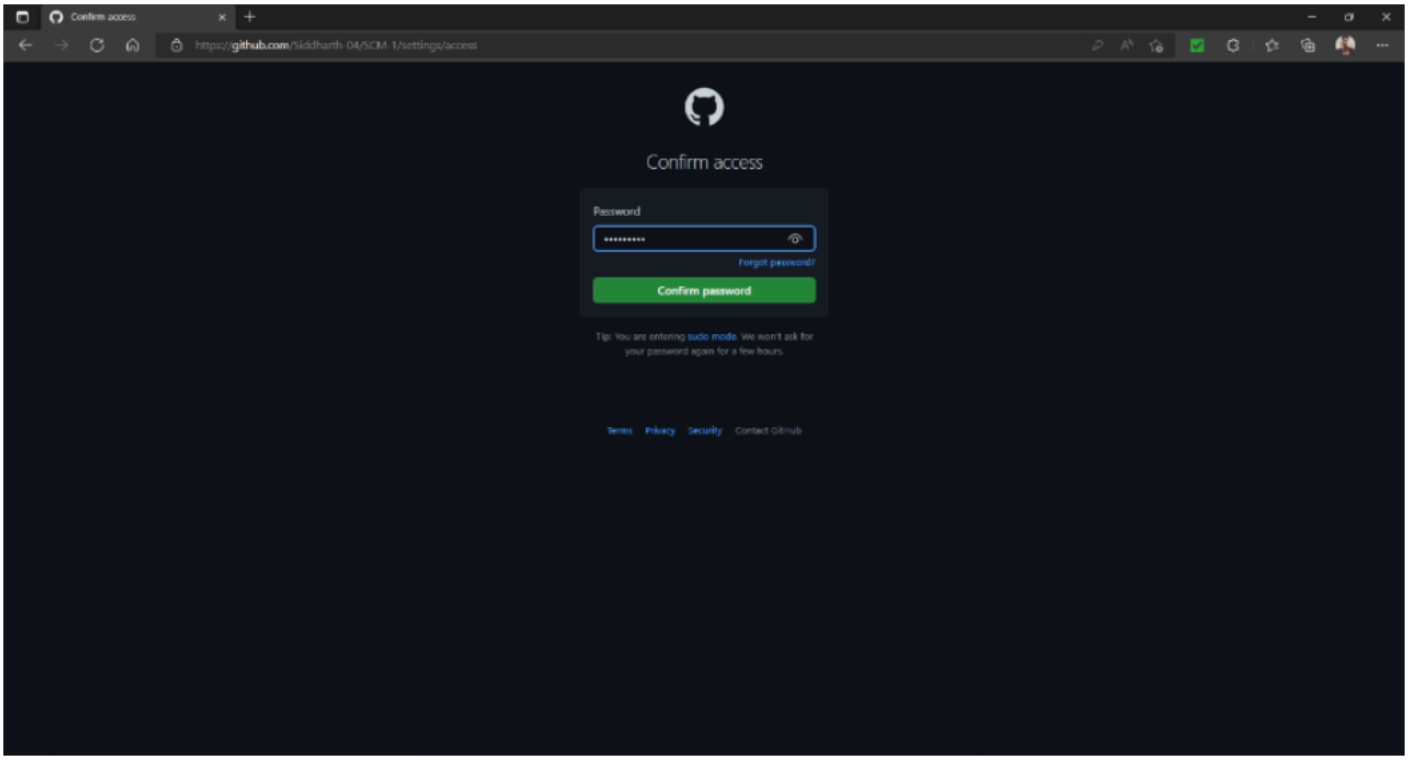
- Now, you have created your repository successfully.
- To add members to your repository, open your repository and select settings option in the navigation bar.



- Click on Collaborators option under the access tab.

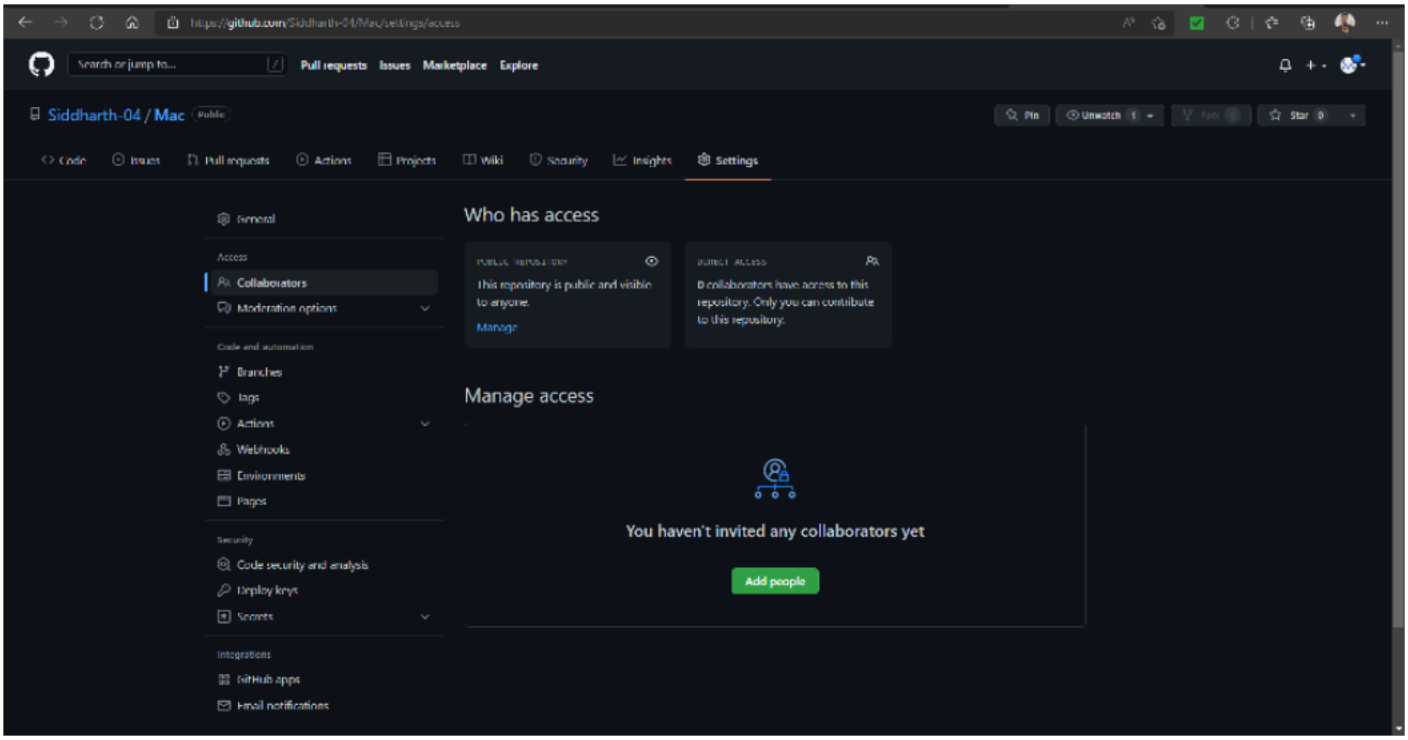


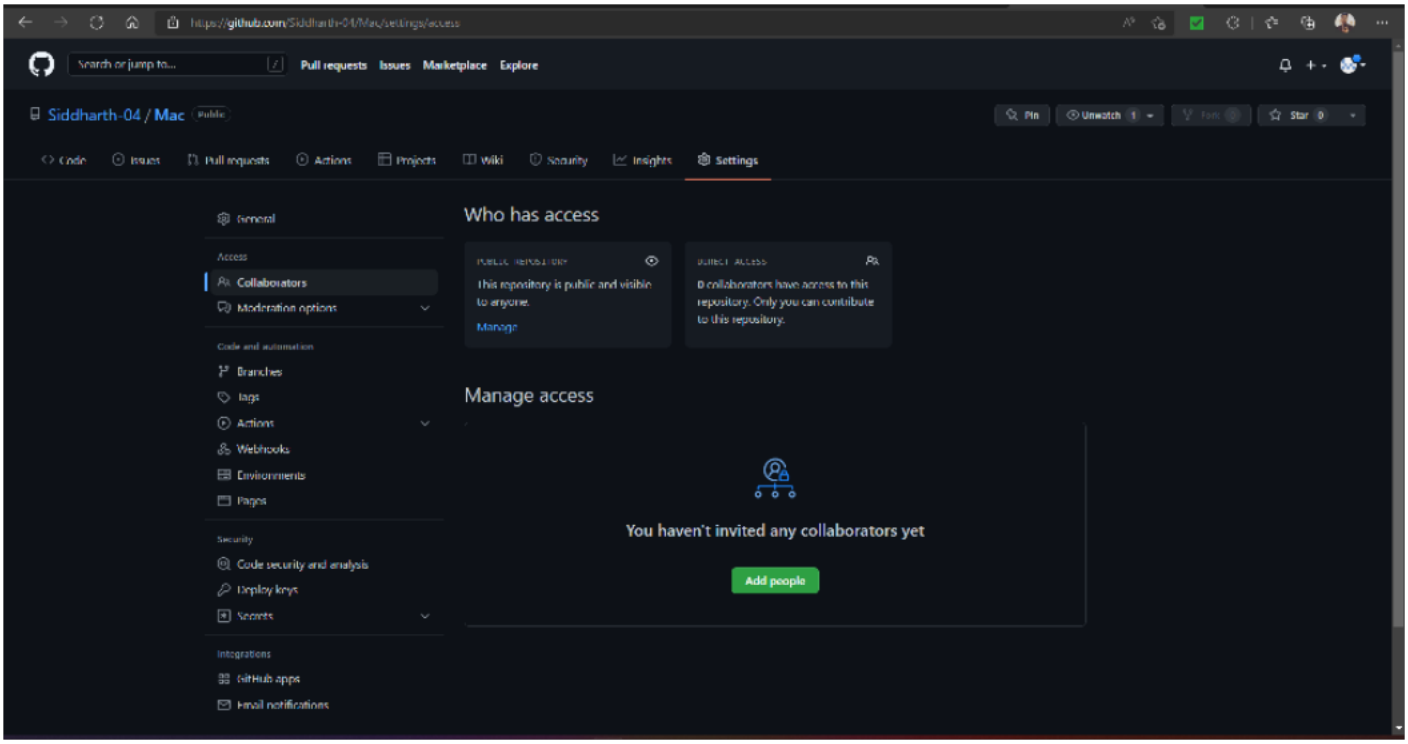
- After clicking on collaborators GitHub asks you to enter your password to confirm the access to the repository.

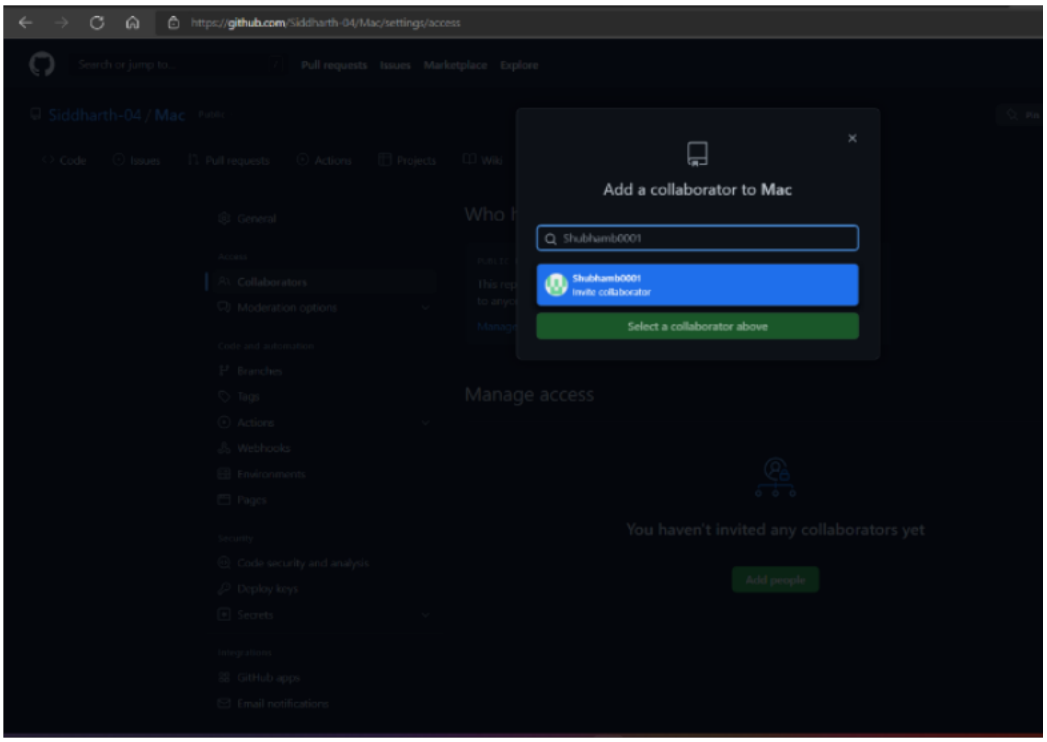


- After entering the password, you can manage access and add/remove team members to your project.

- To add members, click on the add people option and search the id of your respective team member.

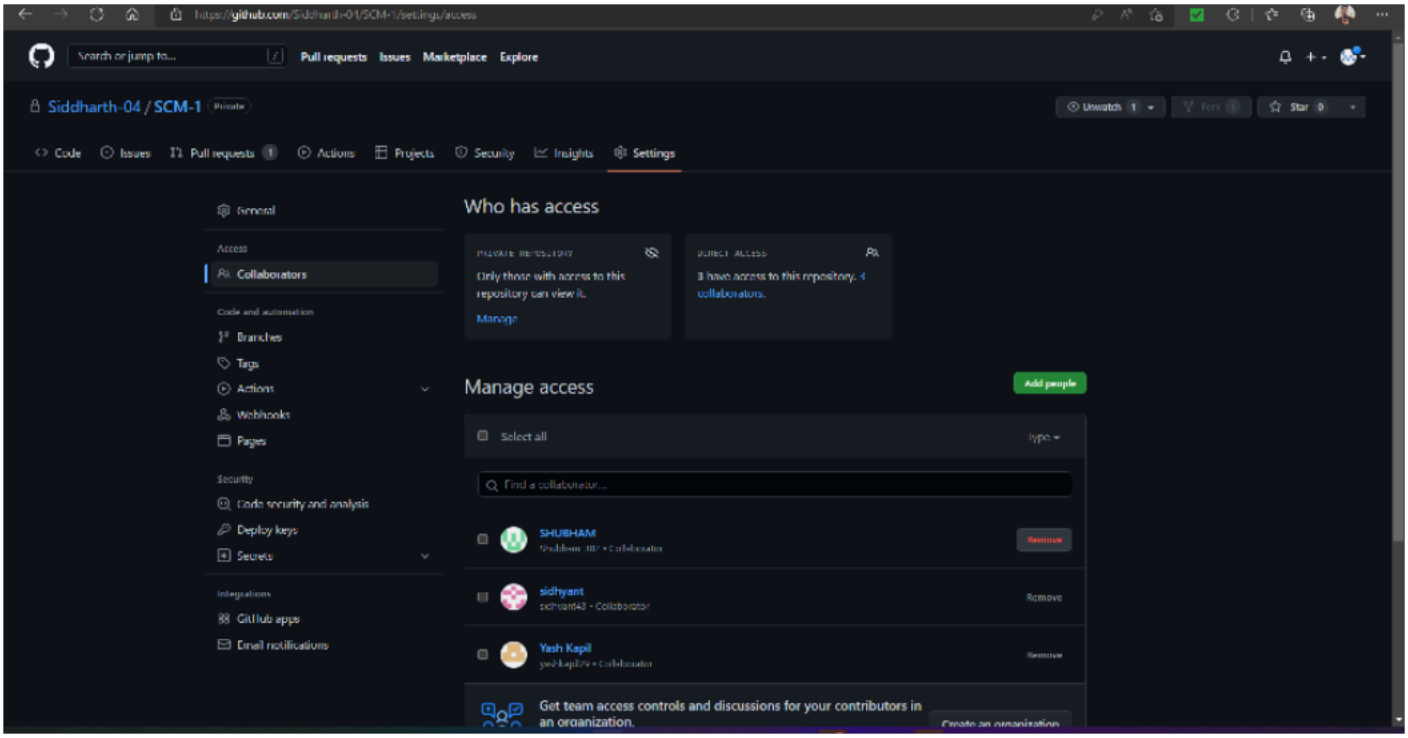




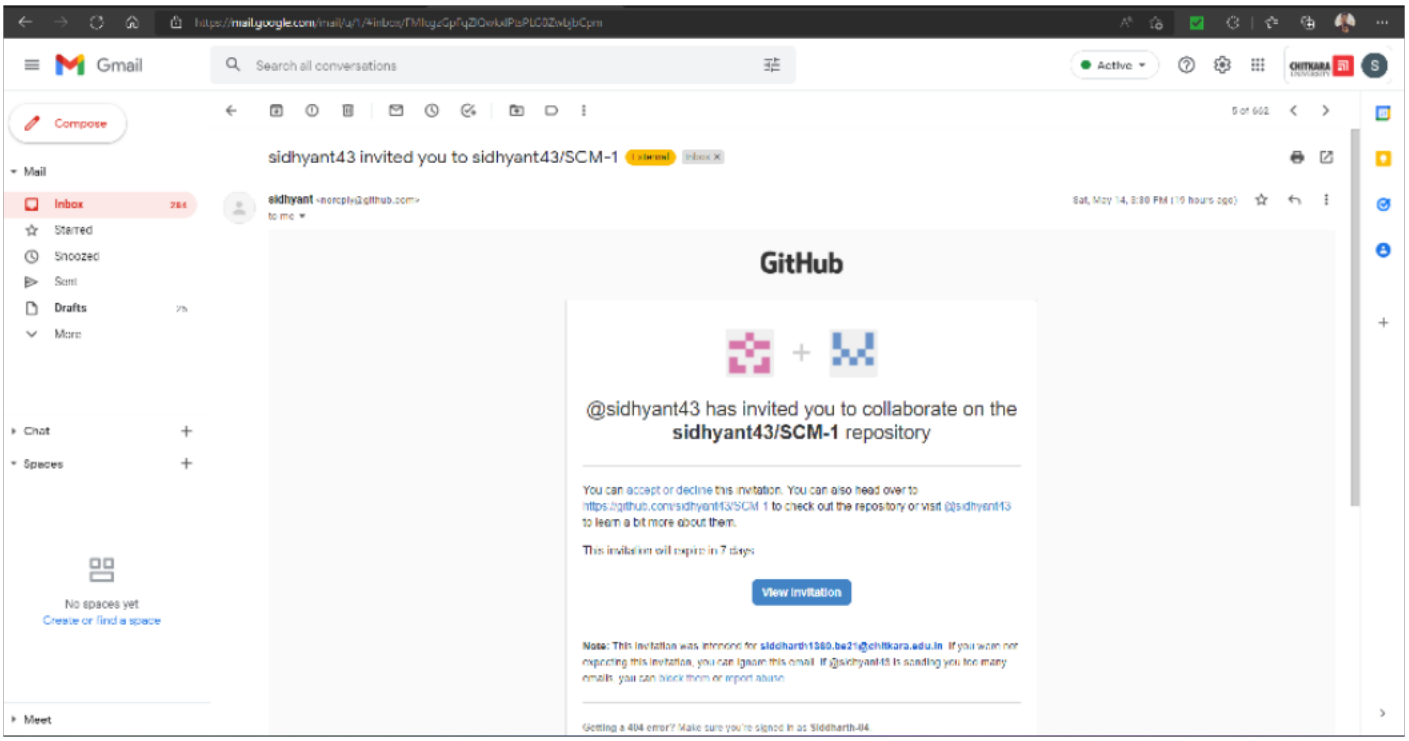


- Now, click on green box to add member in your repository.

- To remove any member, click on remove option available in the last column of member's respective row.



- To accept the invitation from your team member, open your email registered with GitHub.
- You will receive an invitation mail from the repository owner. Open the email and click on accept invitation.
- You will be redirected to GitHub where you can either select to accept or decline the invitation.



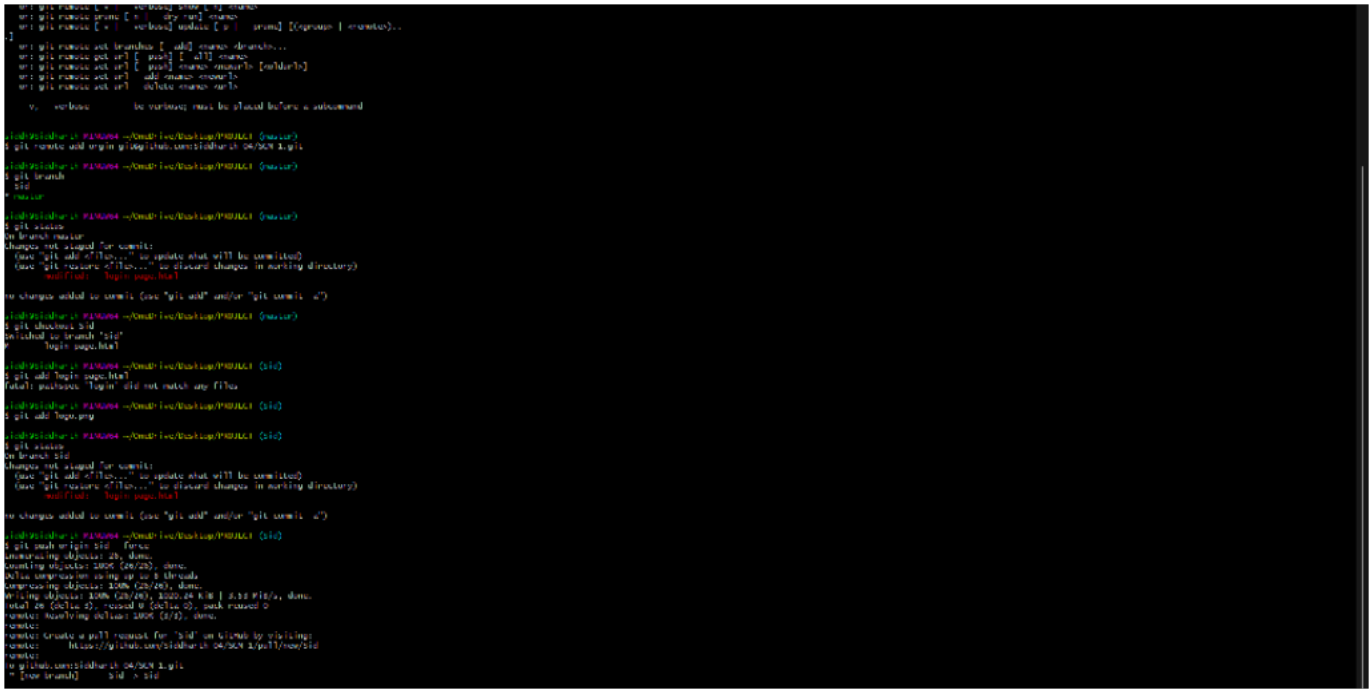
- Now all members are ready to contribute to the project.

Open and Close a Pull Request

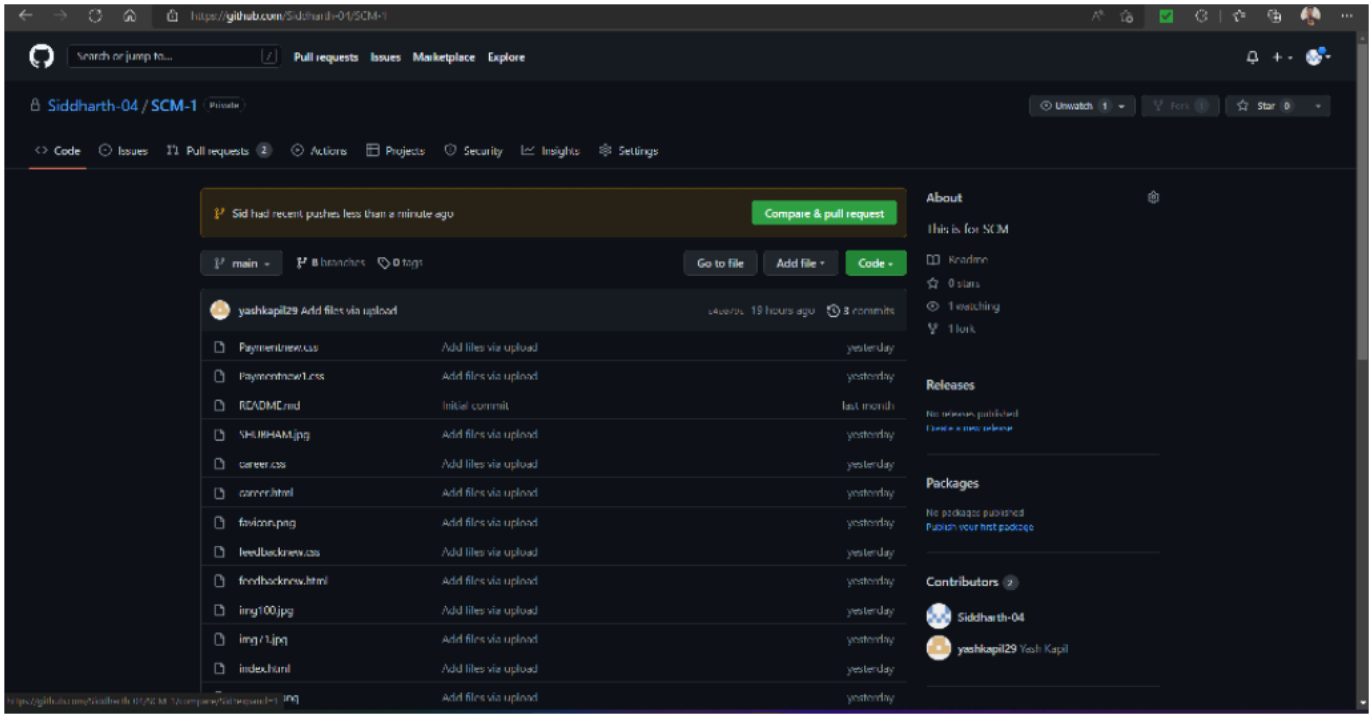
- To open a pull request we first have to make a new branch, by using git branch.
- After making new branch we add a file to the branch or make changes in the existing file.

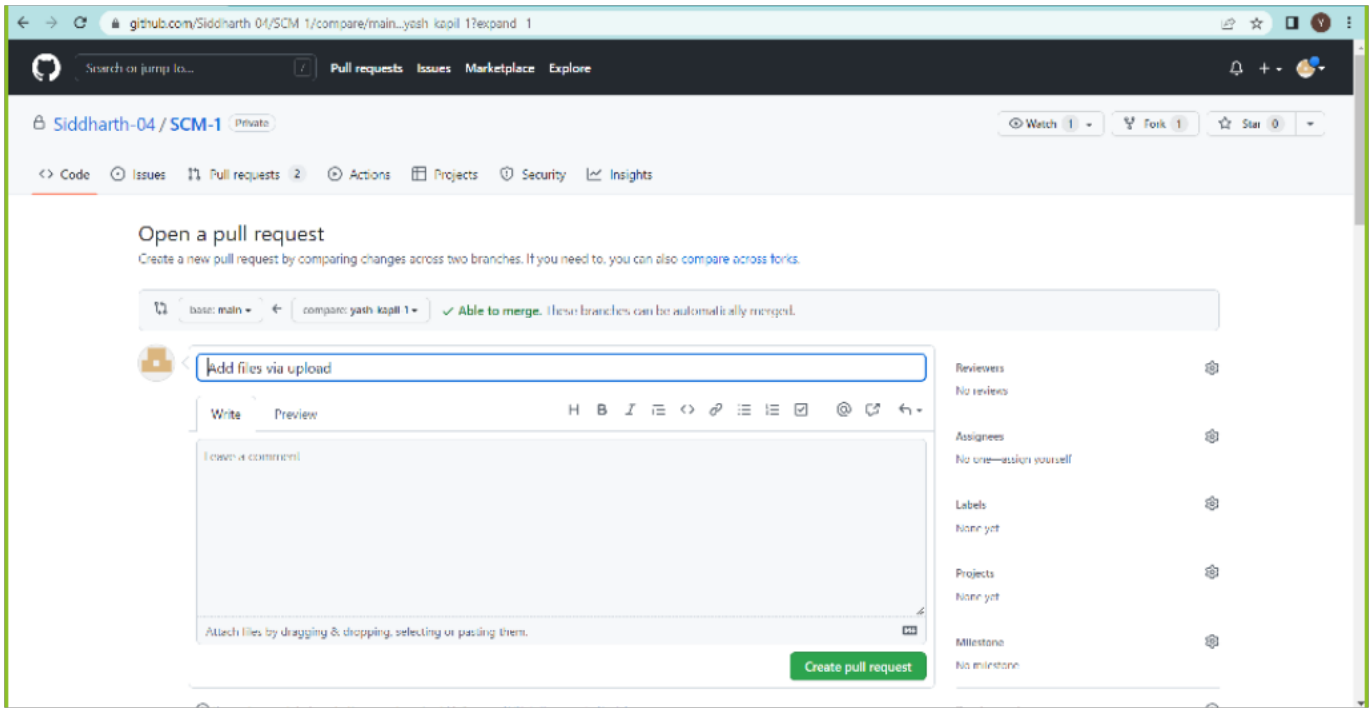
Add and commit the changes to the local repository.

- Use git push origin to push the new branch to the main repository.



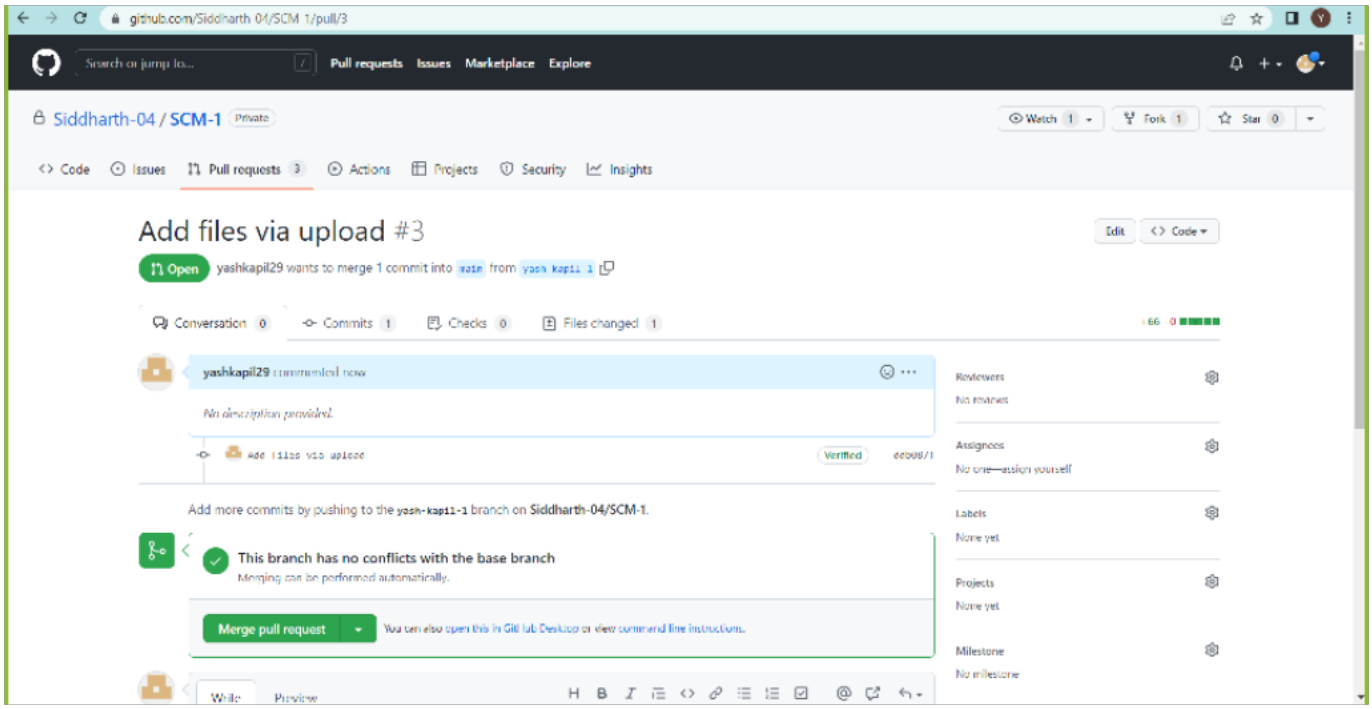
- After pushing new branch GitHub will either automatically ask you to create a pull request or you can create your own pull request by selecting the option compare & pull request.



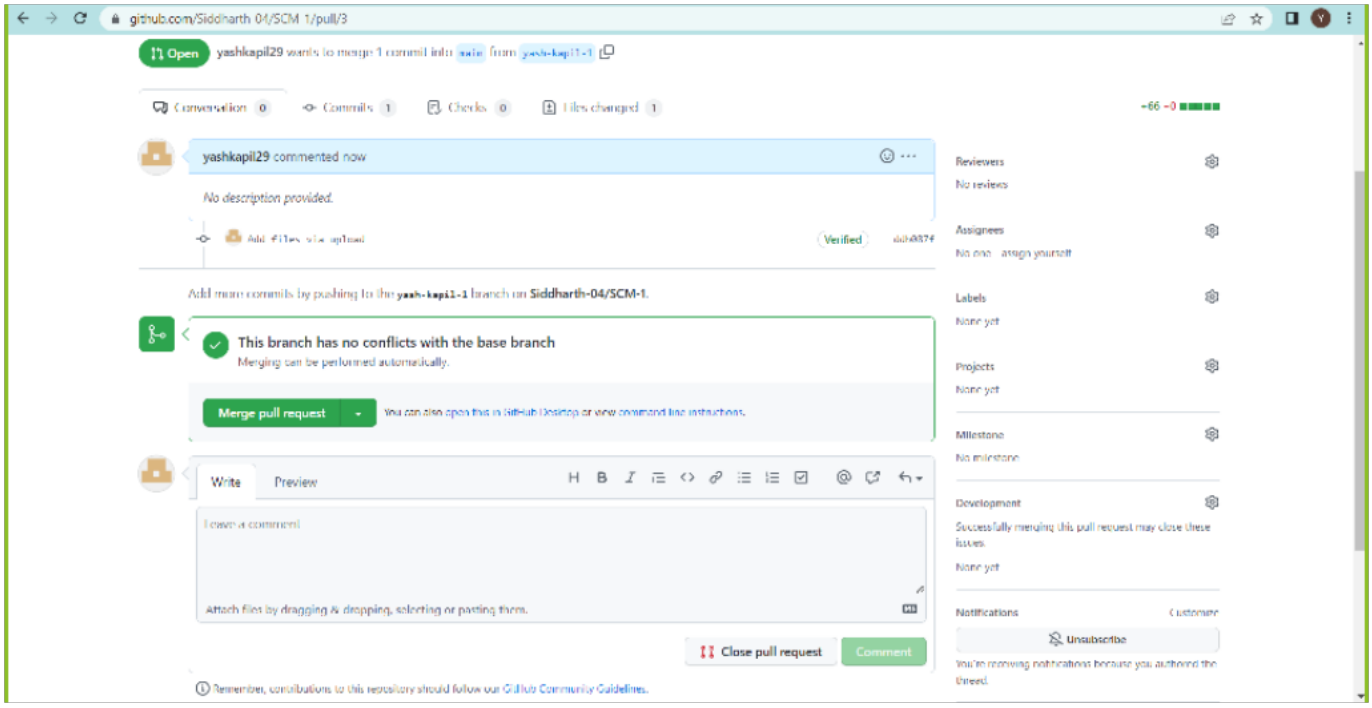


- To create your own pull request, click on create pull request option.

- GitHub will detect any conflicts and ask you to enter a description of your pull request.



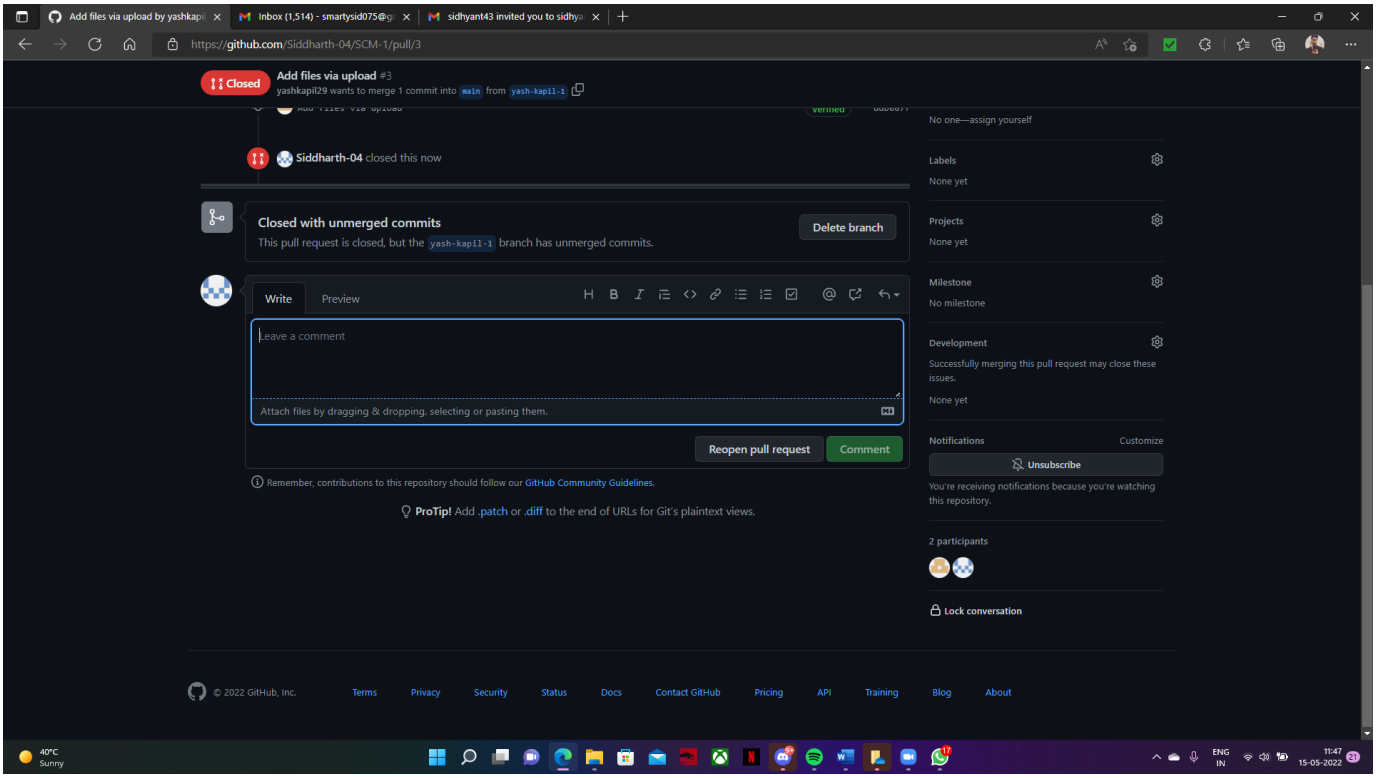
- After opening a pull request all the team members will be sent the request if they want to merge or close the request.



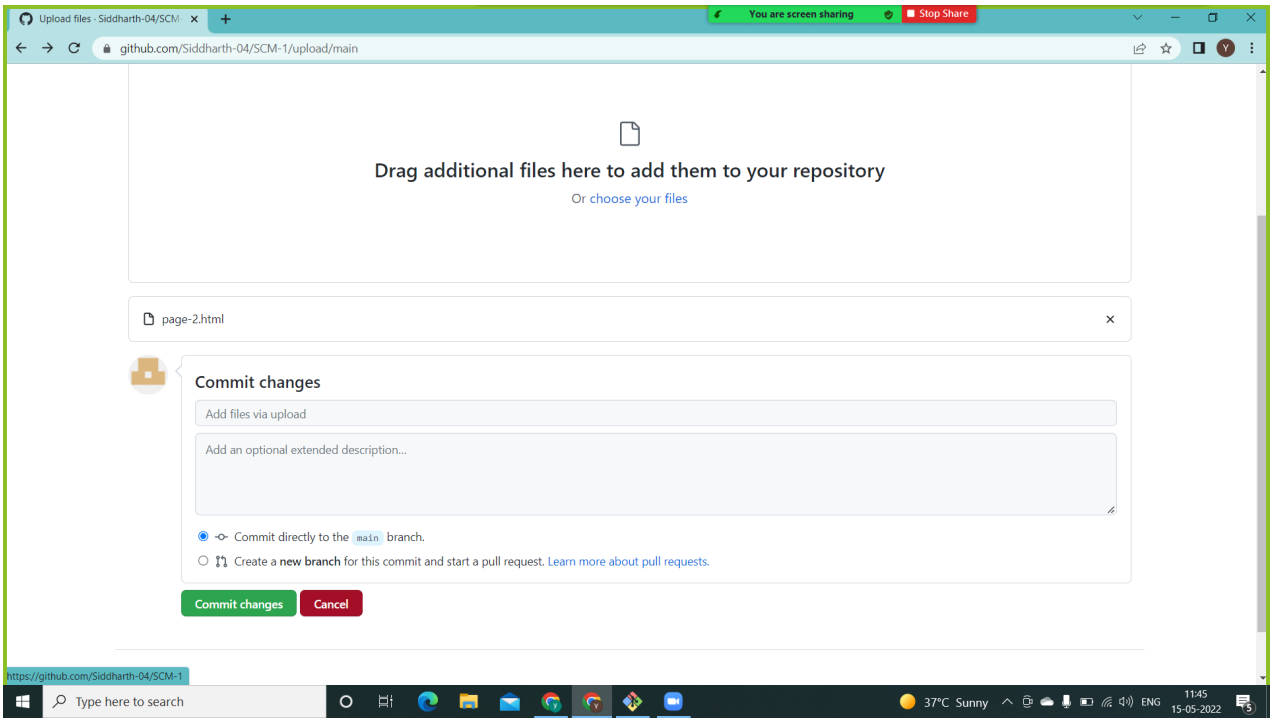
- If the team member chooses not to merge your pull request they will close the pull request.

- To close the pull request simply click on close pull request and add comment/ reason why you closed the pull request.

- You can see all the pull request generated and how they were dealt with by clicking on pull request option.



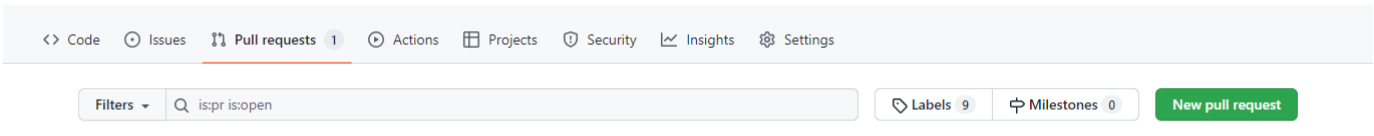
If the Pull request is successfully merged by the team member:



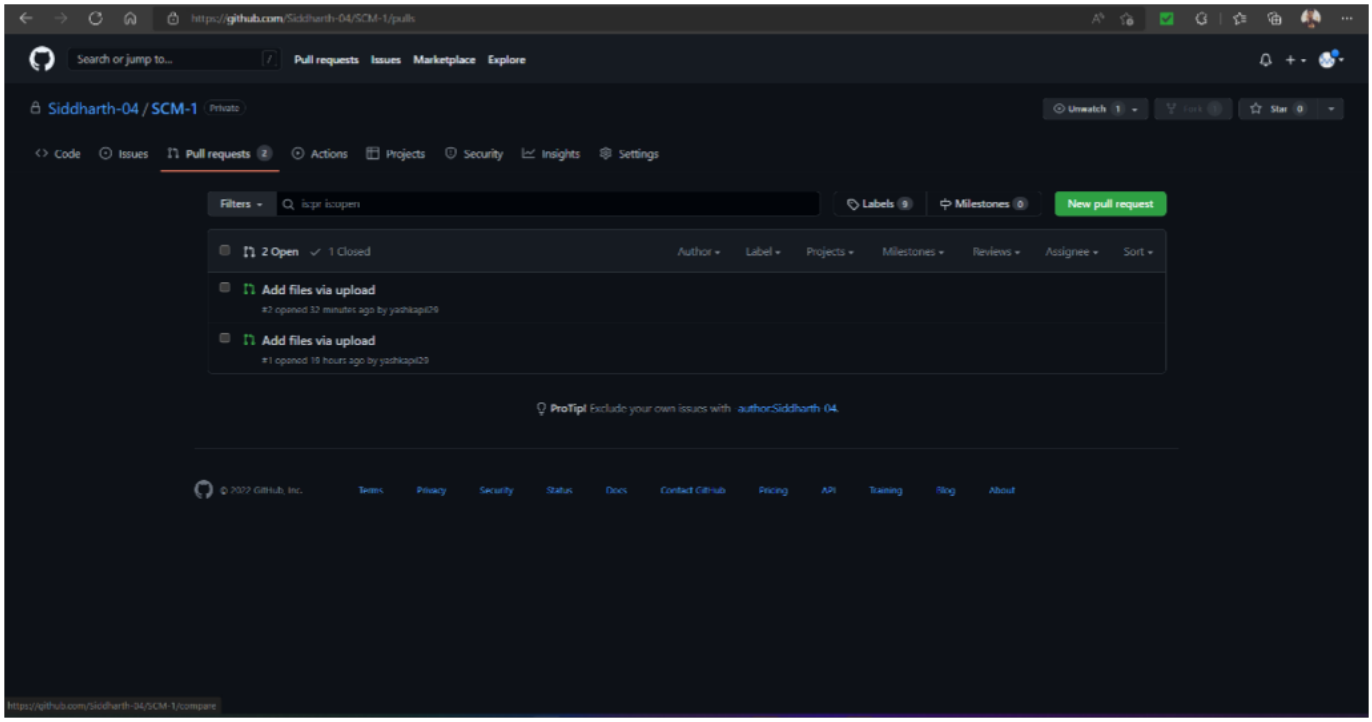
Create a pull request on a team member's repo and close pull requests generated by team members on own Repo as a maintainer

To create a pull request on a team member's repository and close requests by any other team members as a maintainer follow the procedure given below: -

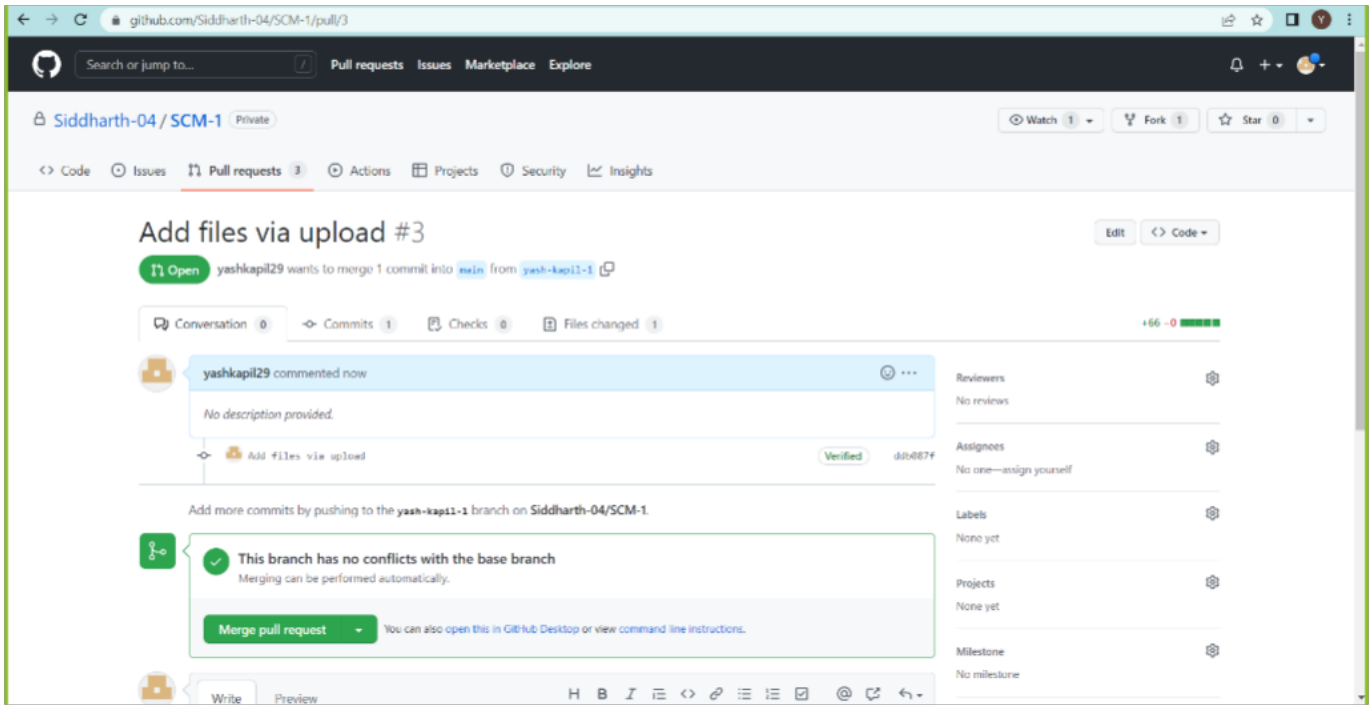
- Do the required changes in the repository, add and commit these changes in the local repository in a new branch.
- Push the modified branch using git push origin branchname.
- Open a pull request by following the procedure from the above experiment.
- The pull request will be created and will be visible to all the team members.
- Ask your team member to login to his/her Github account.
- They will notice a new notification in the pull request menu.



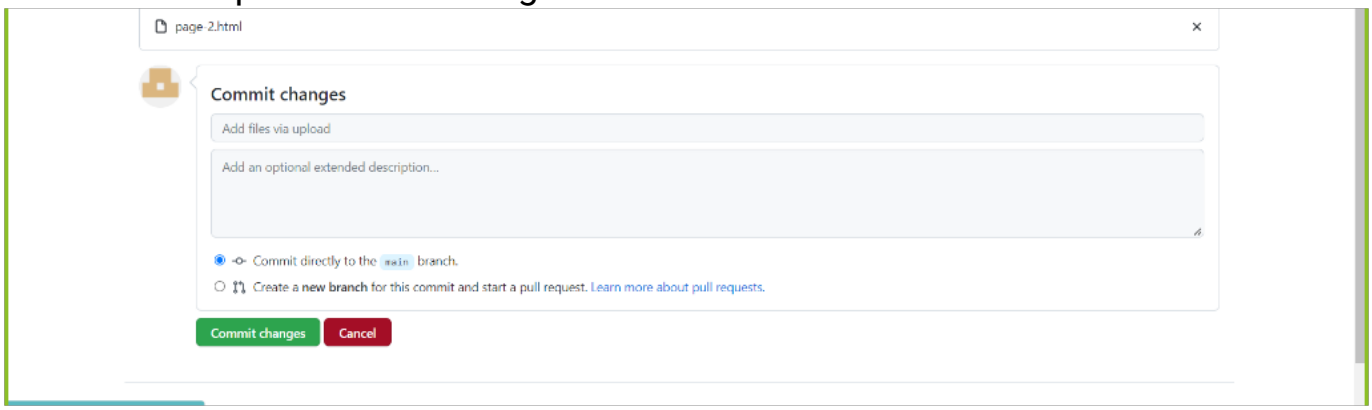
- Click on it. The pull request generated by you will be visible to them.



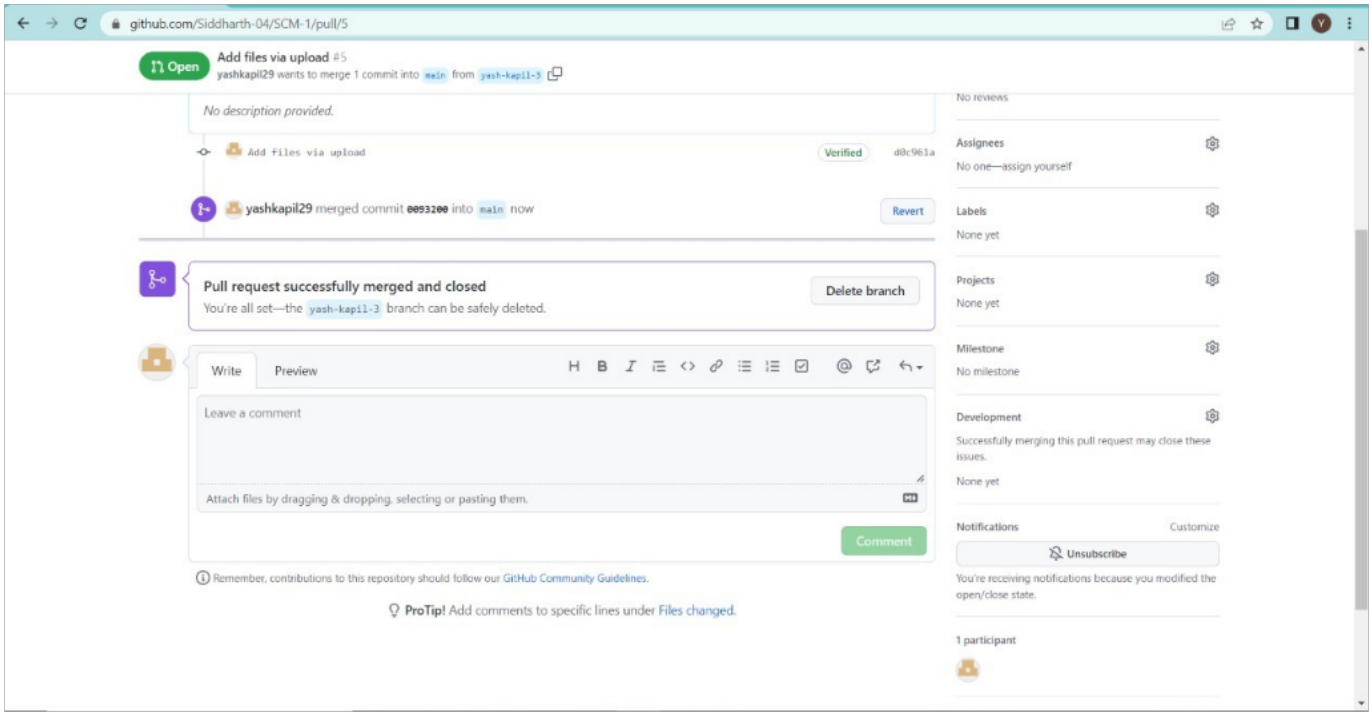
- Click on the pull request. Two options will be available, either to close the pull request or merge the request with the main branch.
- By selecting the merge pull request, the main branch will get updated for all the team members.



- By selecting close the pull request the pull request is not accepted and not merged with main branch.



- The process is similar to closing and merging the pull request by you. It simply includes an external party to execute.
- The result of merging the pull request is shown below.


github.com/Siddharth-04/SCM-1/pull/5
Open
Add files via upload #5
yashkapil29 wants to merge 1 commit into main from yash-kapil-3
No description provided.
Add files via upload
Verified
d0c961a
yashkapil29 merged commit ee53ee into main now
Revert
Pull request successfully merged and closed
You're all set—the yash-kapil-3 branch can be safely deleted.
Delete branch
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Comment
Remember, contributions to this repository should follow our GitHub Community Guidelines.
ProTip! Add comments to specific lines under Files changed.
No reviews
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Development
Successfully merging this pull request may close these issues.
None yet
Notifications
Customize
Unsubscribe
You're receiving notifications because you modified the open/close state.
1 participant


Siddharth-04 closed this 5 minutes ago
None yet
Milestone

6.Workflow and Discussion

Yash Kapil : Added Content

Created Fork

Pull request

Created branch and merged with new branch

Shubham : Created Repository

Added Collaborators

Created New Branches

Siddharth : Uploaded Project

Pushed data in several branches

Worked on Pull request

Sidhyant : Pull data from branch

Merges and close Pull request

Commited changes in Branches

7.Reference

In making this project we have referred to some websites that are metioned below:

1. www.javapoint.com
2. www.ourcodingclub.com
3. www.atlassian.com