

A Project
report on

“Task 2”

with

Source Code Management

(CS181)

Submitted by

Name :-Aashima Mahajan Roll No. :-2110990021

Team Member 1 Name :-Aashish Bhardwaj Roll No. :- 2110990022

Team Member 2 Name :- Aashima Mahajan Roll No. :-2110990021

Team Member 3 Name :-Abhijit Singh Roll No. :- 2110990041



Department of Computer Science & Engineering
Chitkara University Institute of Engineering and Technology, Punjab

Jan- June
(2021-22)

Institute/School: - **Chitkara University Institute of Engineering and Technology**
Name

Department Name: - **Department of Computer Science & Engineering**

Programme Name: - Bachelor of Engineering (B.E.), Computer Science& Engineering

Management

Course Code: - **CS181** Semester/
Batch: **2nd /
2021**
-

Vertical Name: - **Beta** Group No: - **G01**

Course Name: - Source Code Session: - **2021-22**

Course **Dr. Monit Kapoor**
Coordinator:-

Faculty Name: -
Dr.Monit Kapoor

INDEX

| S. NO | Experiment Name | Page No. |
|-------|---|----------|
| 1. | Introduction | 1-6 |
| 2. | Create a distributed Repository and addmembers in project team | 7-11 |
| 3. | Open and close a pull request | 12-15 |
| 4. | Create a pull request on a team members repoand close pull requests generated by team members on own Repo as a maintainer | 16-18 |
| 5. | Publish and print network graphs | 19-22 |

Introduction

What is GIT and why is it used?

Git is a version control system that is widely used in the programming world. It is used for tracking changes in the source code during software development. It was developed in 2005 by Linus Torvalds, the creator of the Linux operating system kernel.

Git is a speedy and efficient distributed [VCS](#) tool that can handle projects of any size, from small to very large ones. Git provides cheap local branching, convenient staging areas, and multiple workflows. It is free, open-source software that lowers the cost because developers can use Git without paying money. It provides support for non-linear development. Git enables multiple developers or teams to work separately without having an impact on the work of others.

Git is an example of a distributed version control system (DVCS) (hence Distributed Version Control System).

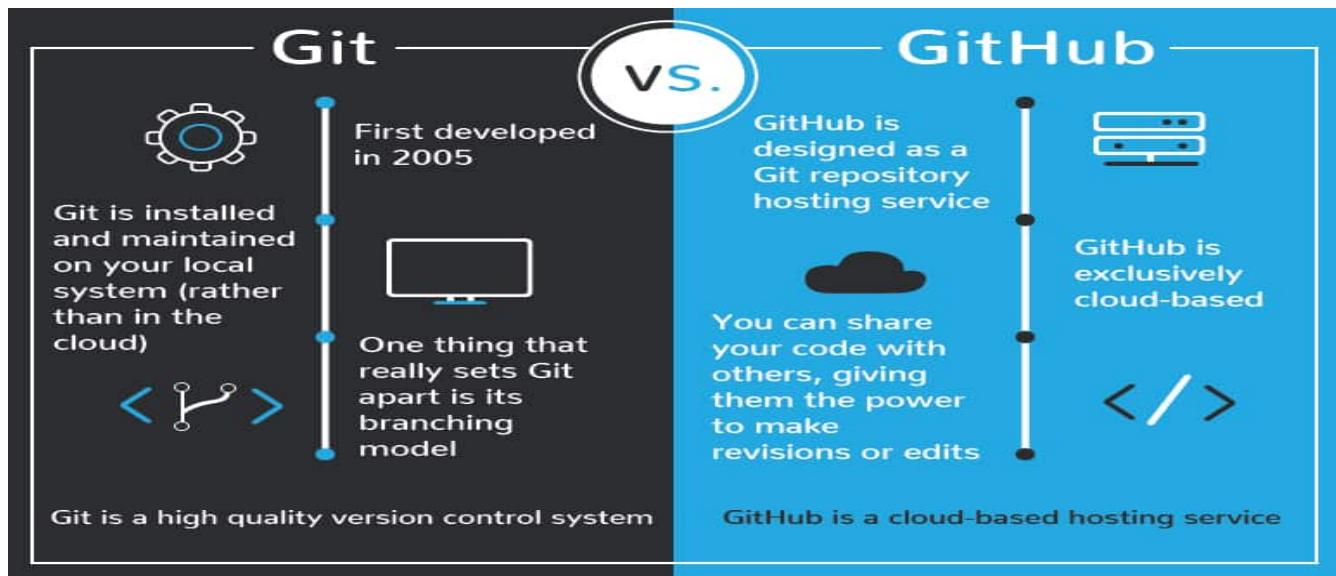


What is GITHUB?

It is the world's largest open-source software developer community platform where the users upload their projects using the software Git.



What is the difference between GIT and GITHUB?



What is Repository?

A repository is a directory or storage space where your projects can live. Sometimes GitHub users shorten this to “repo.” It can be local to a folder on your computer, or it can be a storage space on GitHub or another online host. You can keep code files, text files, image files, you name it, inside a repository.

What is Version Control System (VCS)?

A version control system is a tool that helps you manage “versions” of your code or changes to your code while working with a team over remote distances. Version control keeps track of every modification in a special kind of database that is accessible to the version control software. Version control software (VCS) helps you revert back to an older version just in case a bug or issue is introduced to the system or fixing a mistake without disrupting the work of other team members.

Types of VCS

1. Local Version Control System
2. Centralized Version Control System
3. Distributed Version Control System

- I. **Local Version Control System:** Local Version Control System is located in your local machine. If the local machine crashes, it would not be possible to retrieve the files, and all the information will be lost. If anything happens to a single version, all the versions made after that will be lost.
- AI. **Centralized Version Control System:** In the Centralized Version Control Systems, there will be a single central server that contains all the files related to the project, and many collaborators checkout files from this single server (you will only have a working copy). The problem with the Centralized

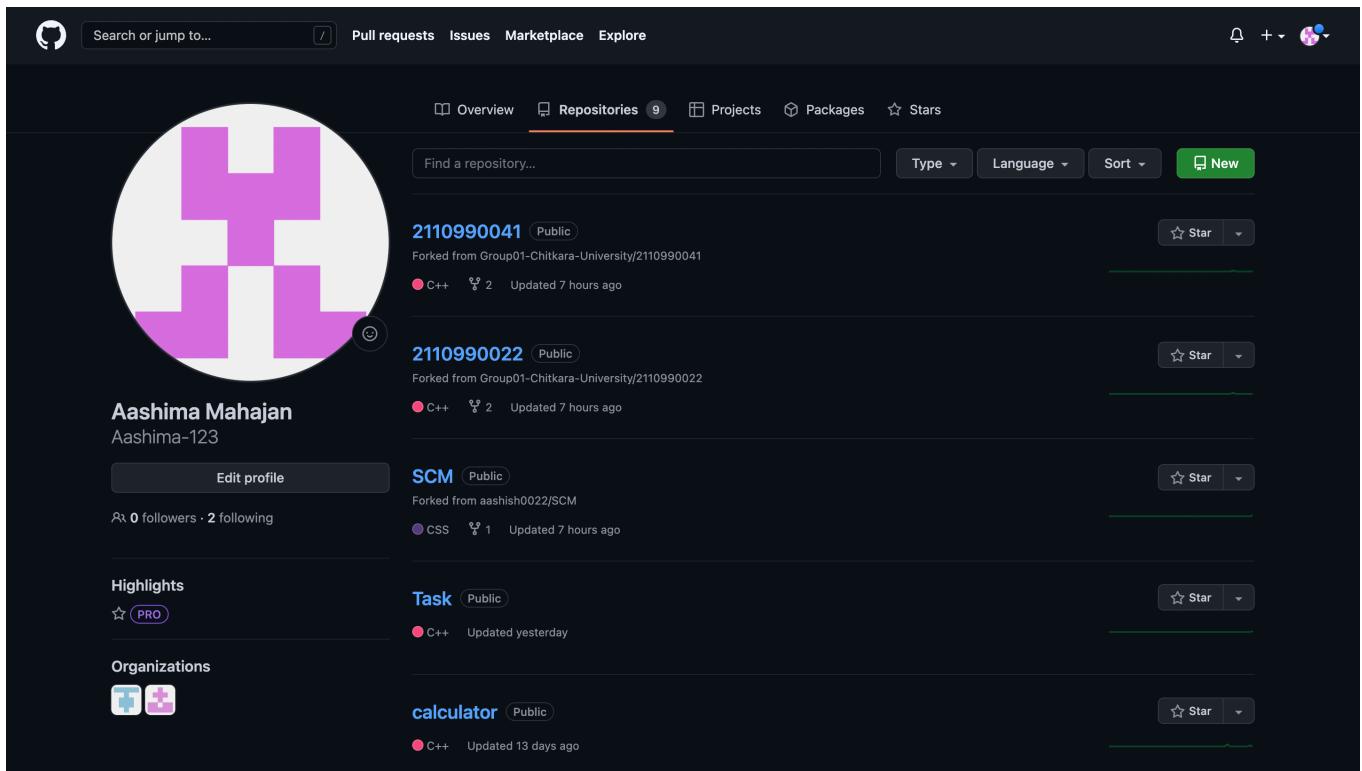
Version Control Systems is if the central server crashes, almost everything related to the project will be lost.

- BI. **Distributed Version Control System:** In a distributed version control system, there will be one or more servers and many collaborators similar to the centralized system. But the difference is, not only do they check out the latest version, but each collaborator will have an exact copy of the main repository on their local machines. Each user has their own repository and a working copy. This is very useful because even if the server crashes we would not lose everything as several copies are residing in several other computers.

Experiment No. 01

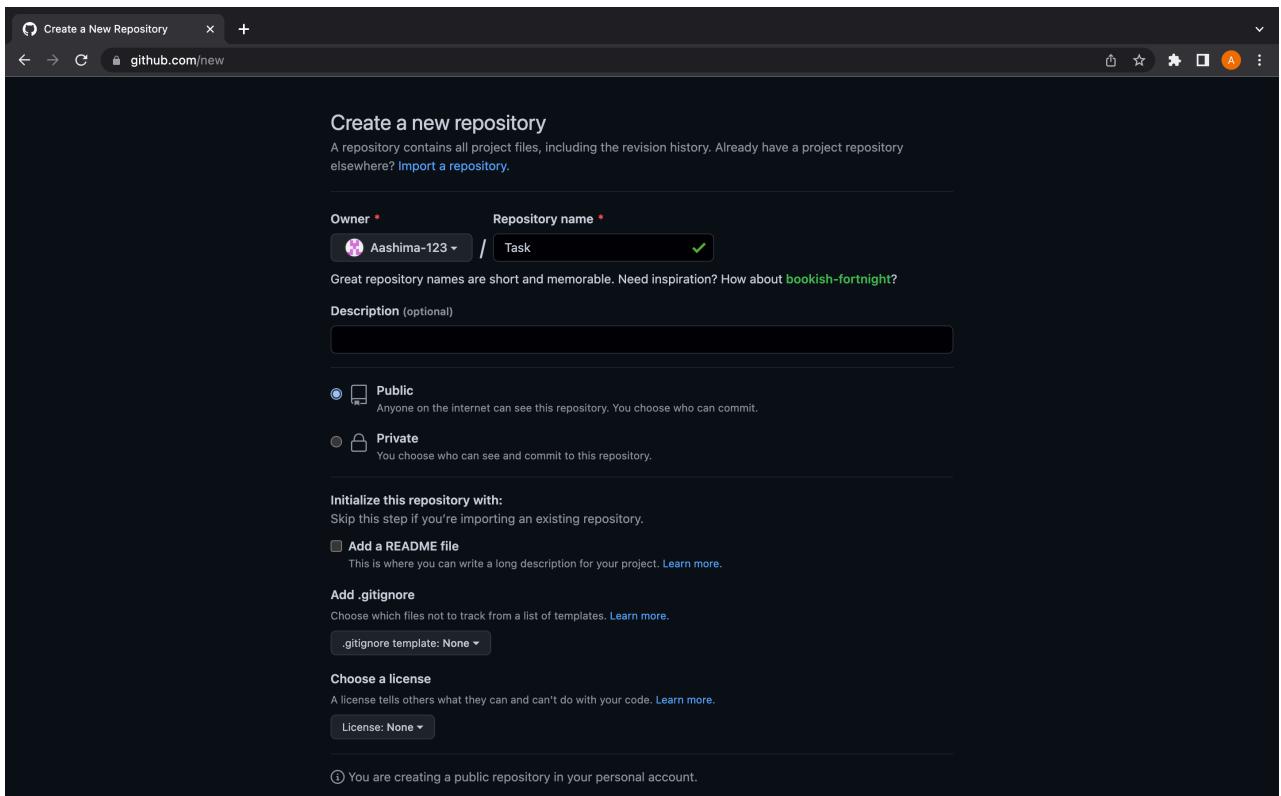
Aim: Create a distributed Repository and add members in project team

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



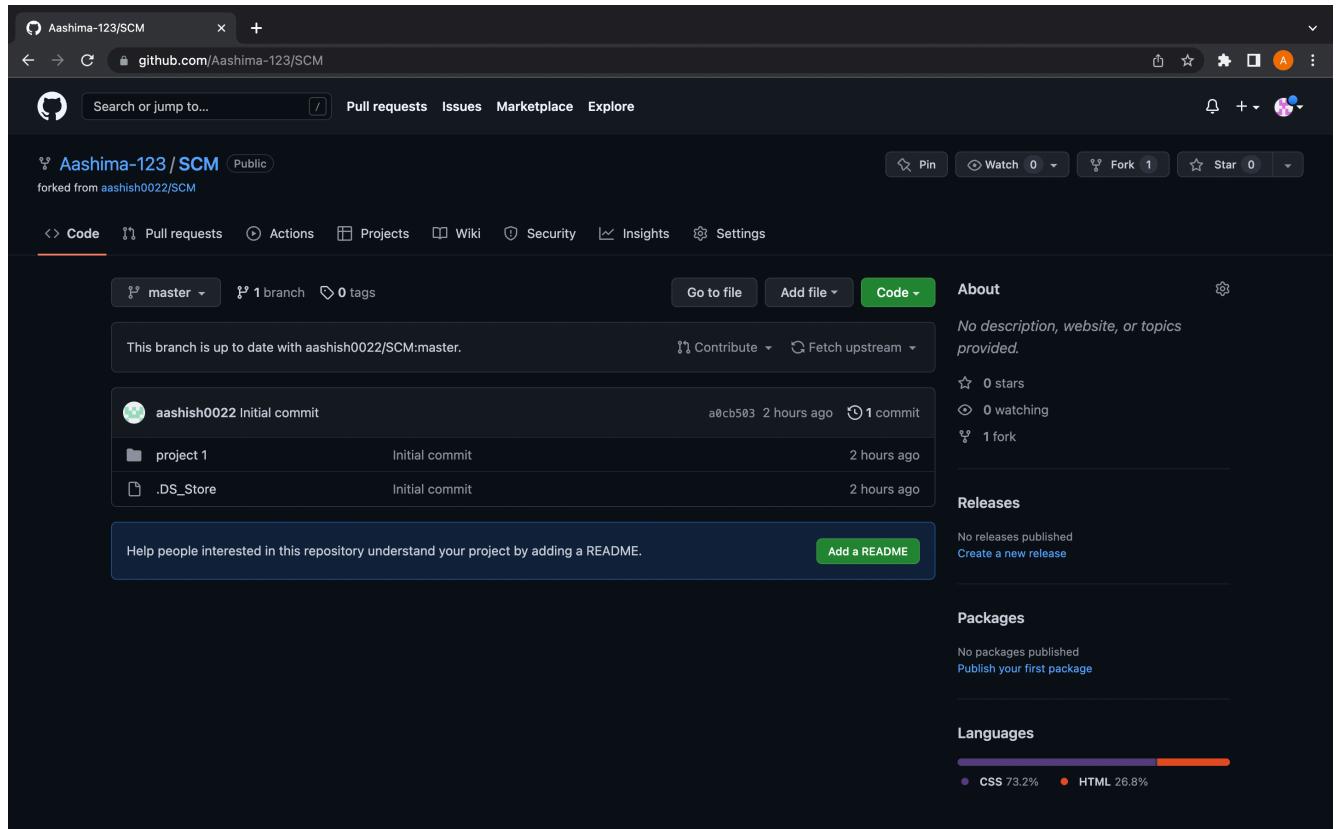
- Click on the 'New' button in the top right corner.
- Enter the Repository name and add the description of the repository.

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



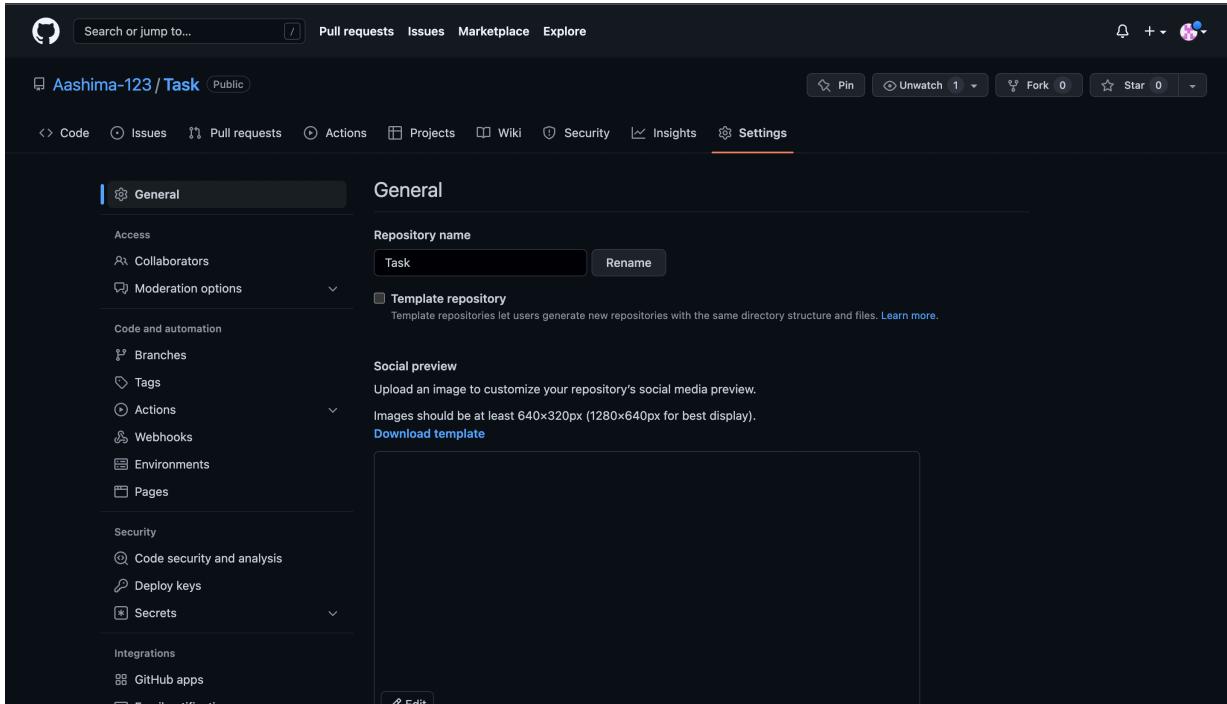
- If you want to import code from an existing repository select the import code option.
- To create a new file or upload an existing file into your repository select the option in the following box.

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

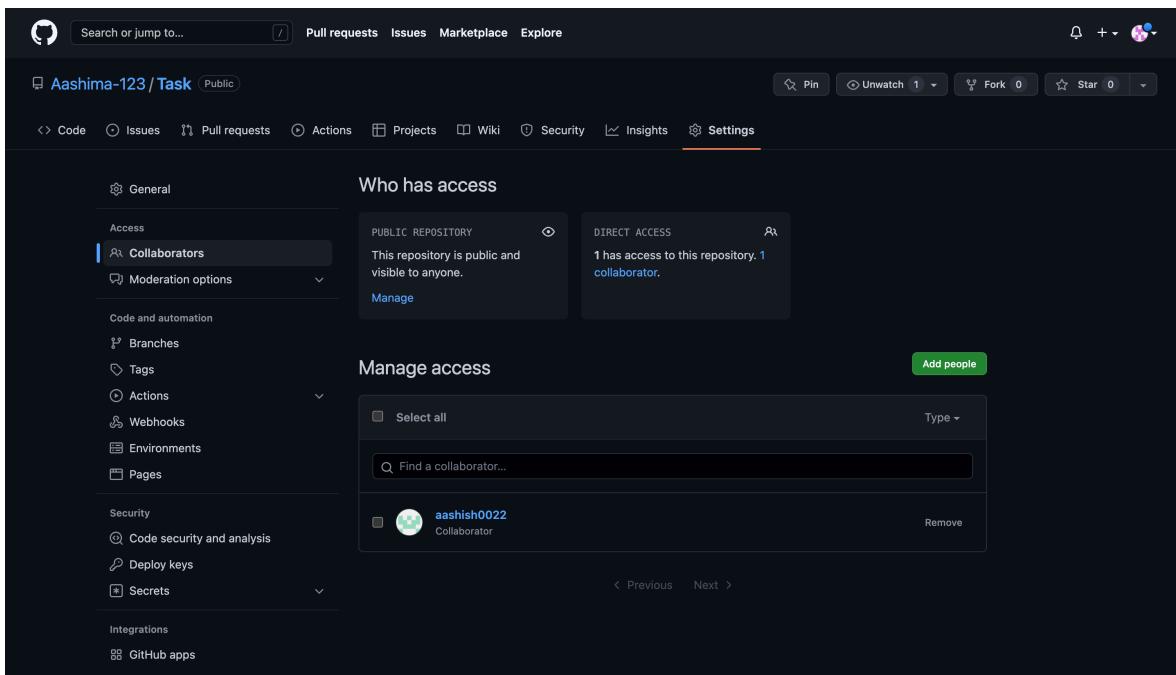


- Click on Collaborators option under the access tab.
- After clicking on collaborators Github asks you to enter your password to confirm the access to the repository.

- 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.

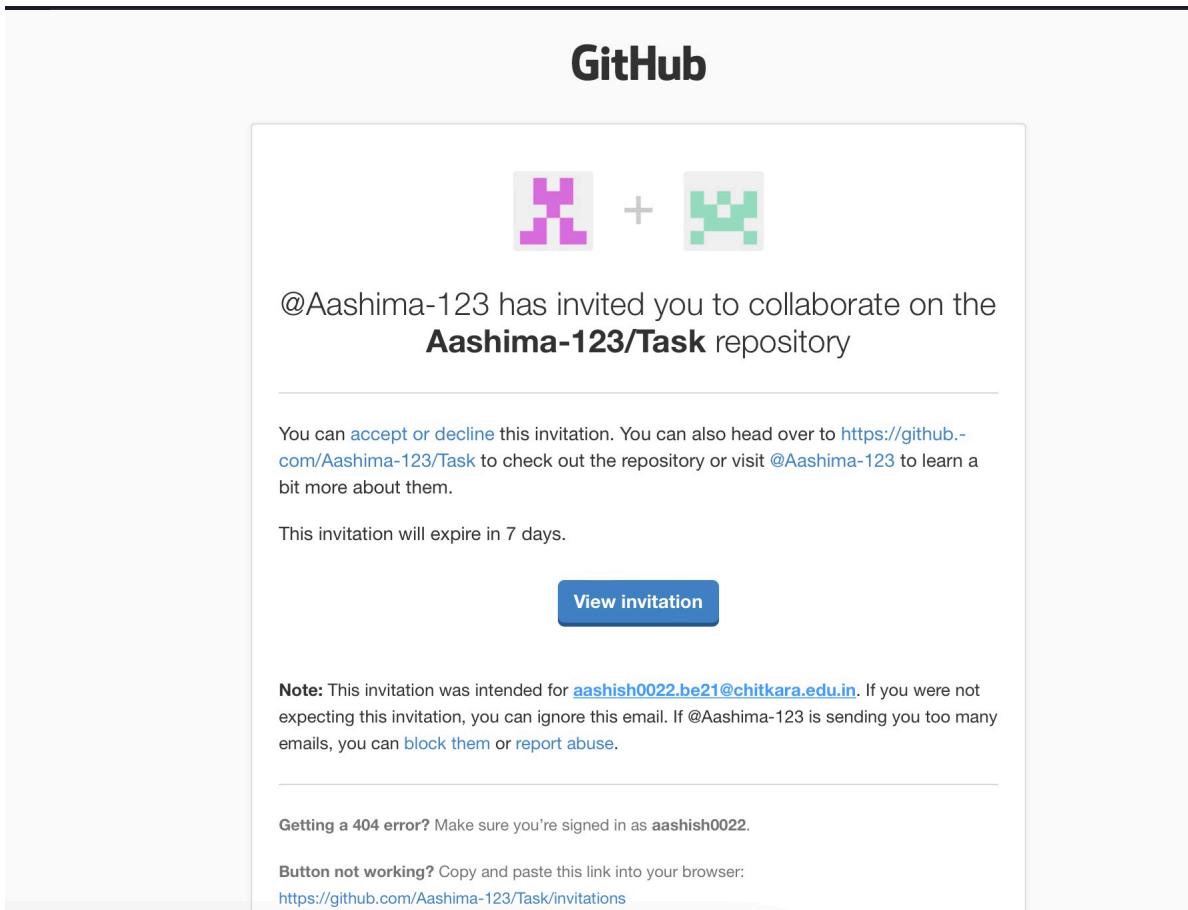


- To remove any member click on remove option available in the last column of member's respective row.



- 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.



- You will be shown the option that you are now allowed to push.
- Now all members are ready to contribute to the project.

Experiment No. 02

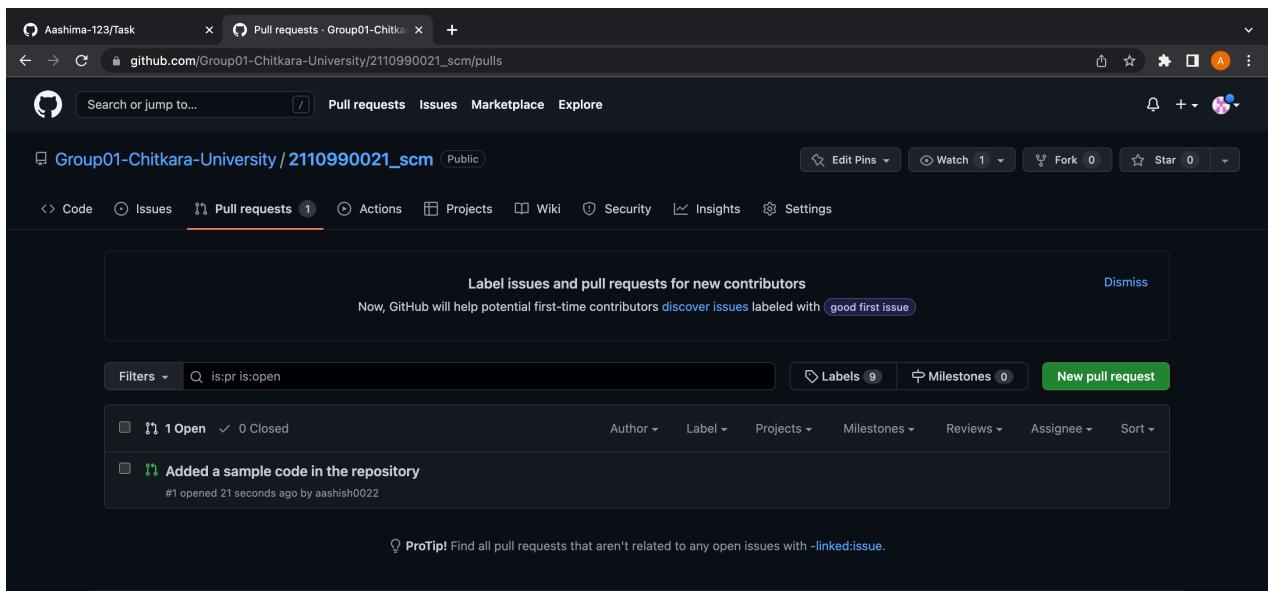
Aim: Open And Close a Pull Request

- To open a pull request we first have to make a new branch, by using gitbranch *branchname* option.
- After making new branch we add a file to the branch or make changes inthe existing file.
- Add and commit the changes to the local repository.

```
Last login: Thu Jun  2 22:15:06 on ttys000
[(base) aashima@aashimas-MacBook-Air Task % git branch
* master
  sample
[(base) aashima@aashimas-MacBook-Air Task % git checkout sample
M      function.cpp
Switched to branch 'sample'
[(base) aashima@aashimas-MacBook-Air Task % touch hello.txt
(base) aashima@aashimas-MacBook-Air Task % ]
```

- Use `git push origin branchname` option 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.

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

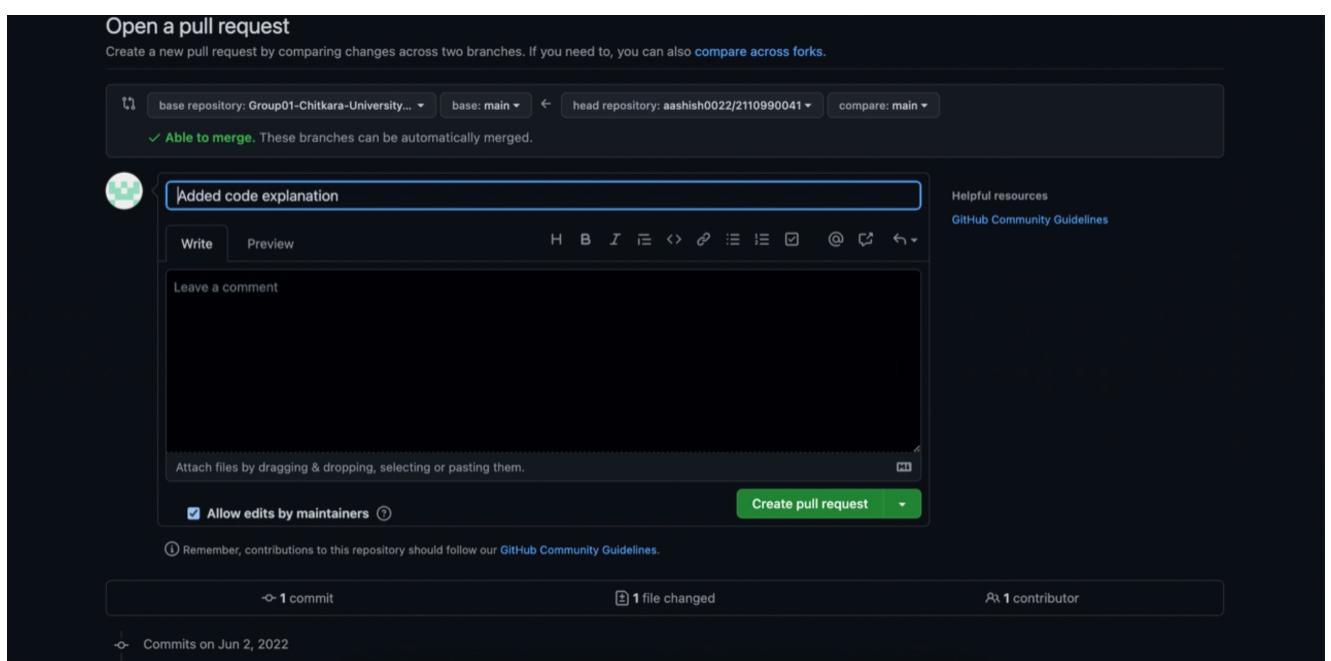


The screenshot shows the GitHub interface for a repository named 'Group01-Chitkara-University / 2110990021_scm'. The 'Pull requests' tab is selected, showing 1 open pull request. The pull request details are as follows:

- Title:** Added a sample code in the repository
- Status:** 1 Open, 0 Closed
- Author:** aashish0022
- Created:** #1 opened 21 seconds ago
- Description:** Added code explanation

Below the pull request, there is a note: "ProTip! Find all pull requests that aren't related to any open issues with -linked:issue."

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



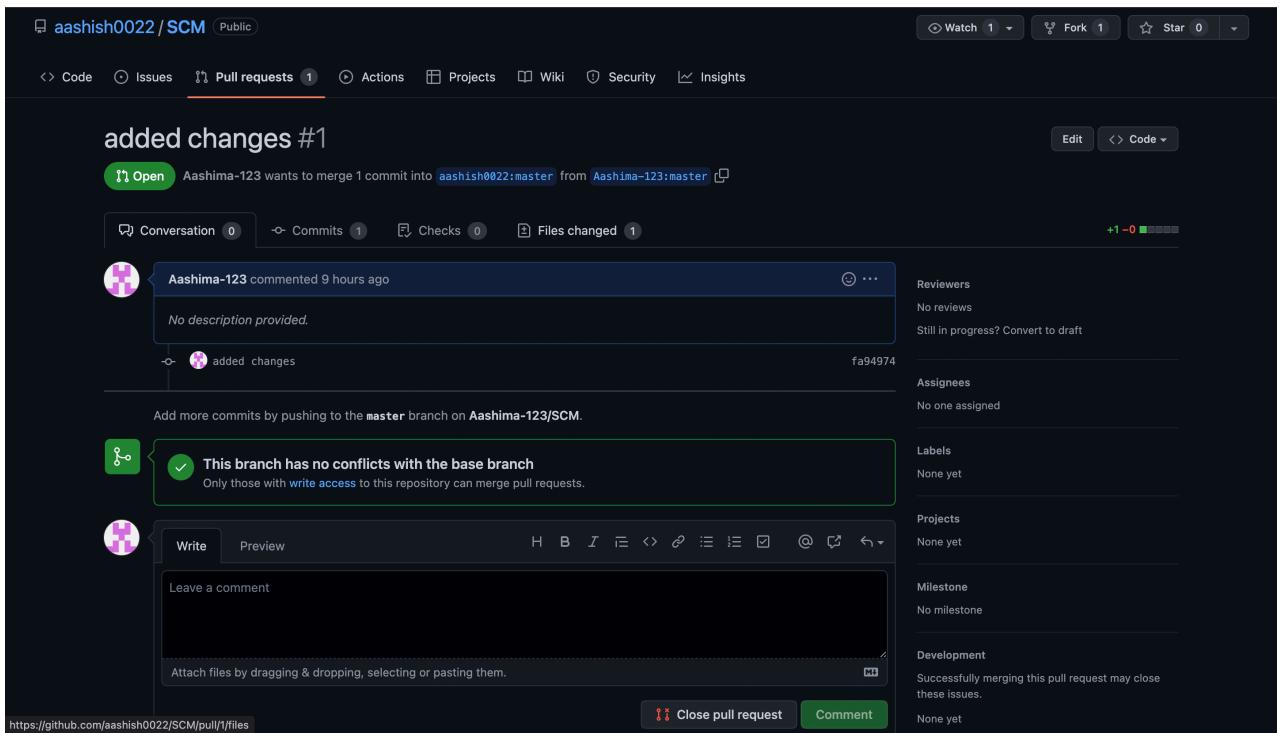
The screenshot shows the 'Open a pull request' form. The base repository is set to 'Group01-Chitkara-University...' and the head repository is 'aashish0022/2110990041'. The 'base: main' dropdown is selected. The 'compare: main' dropdown is also visible. A green checkmark indicates that the branches are 'Able to merge'.

The main area contains a text input field with the placeholder 'Added code explanation'. Below it is a rich text editor toolbar. A comment section with the placeholder 'Leave a comment' is present. At the bottom, there is a 'Create pull request' button and a note about contribution guidelines.

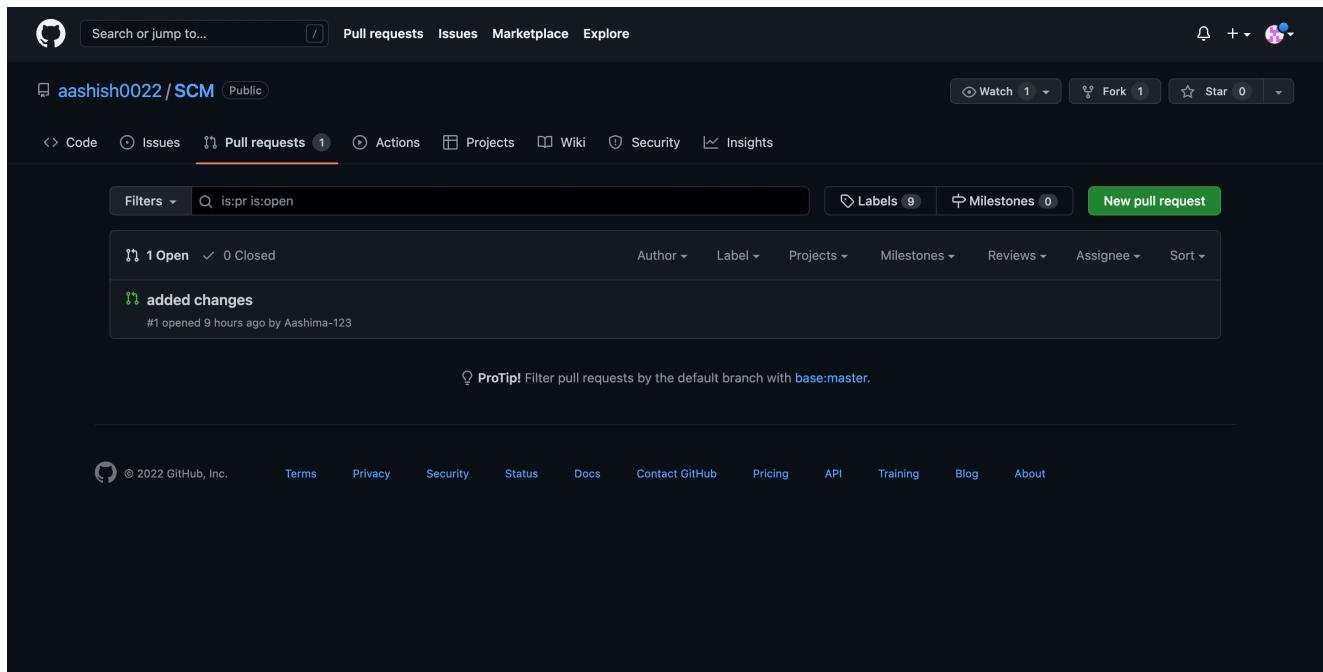
At the bottom of the form, there are summary statistics: 1 commit, 1 file changed, and 1 contributor. A note at the bottom left says 'Commits on Jun 2, 2022'.

- After opening a pull request all the team members will be sent therequest if they want to merge or close the request.
- If the team member chooses not to merge your pull request they willclose you're the pull request.

- To close the pull request simply click on close pull request and addcomment/ 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.

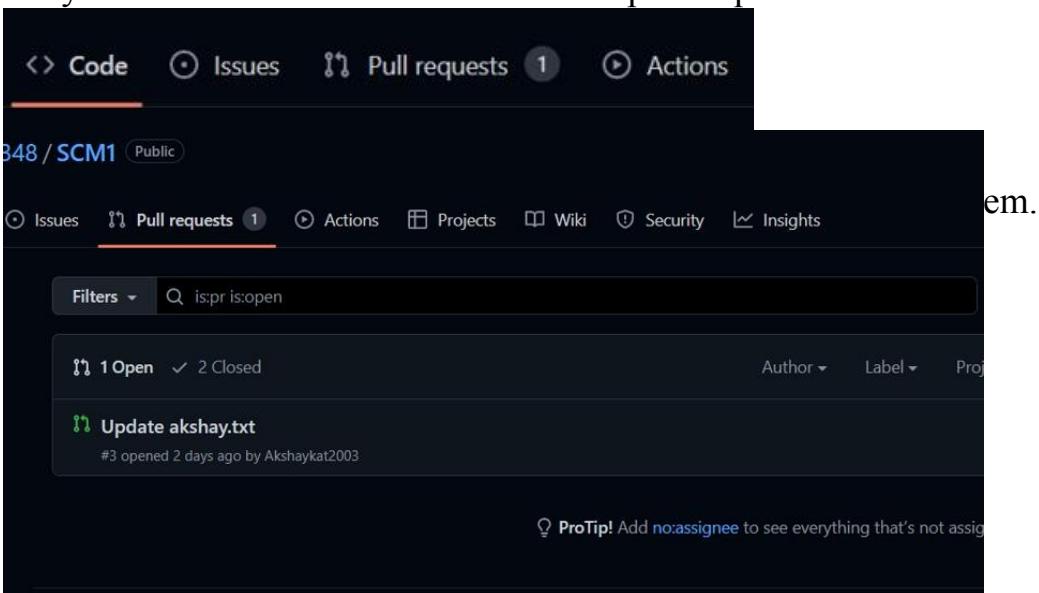


Experiment No. 03

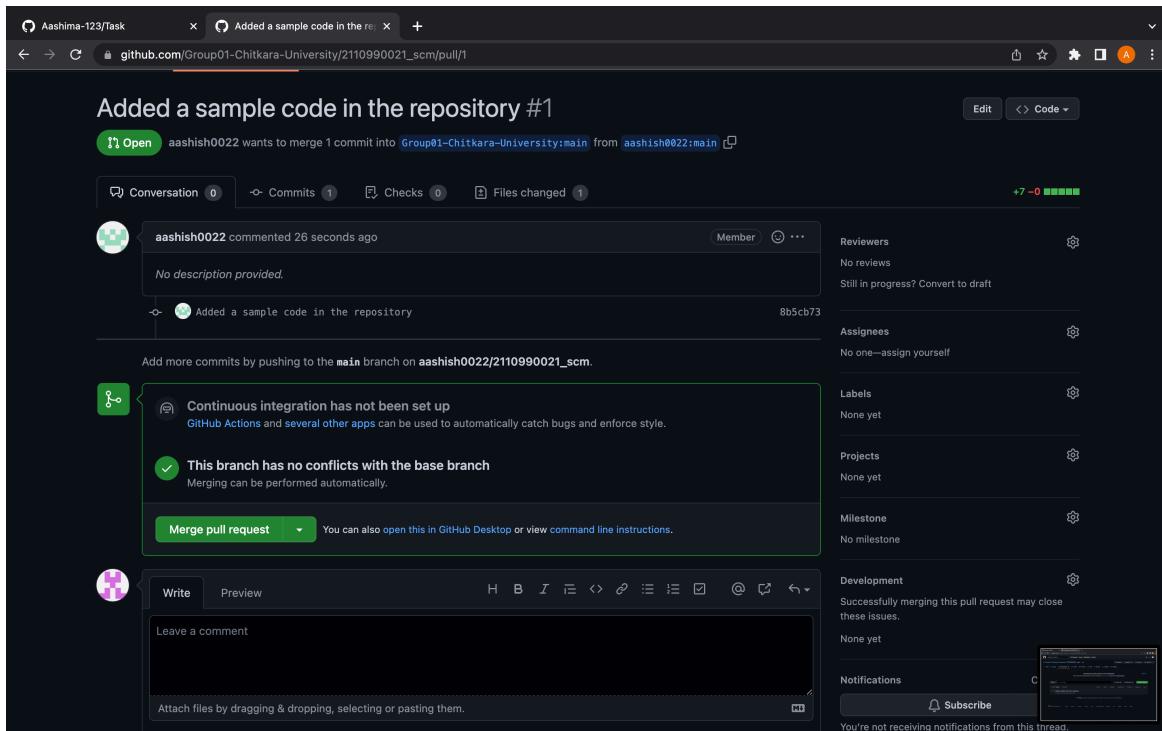
Aim: Create a pull request on a team member's repo and close pull requests generated by team members on own Repo as a maintainer

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:-

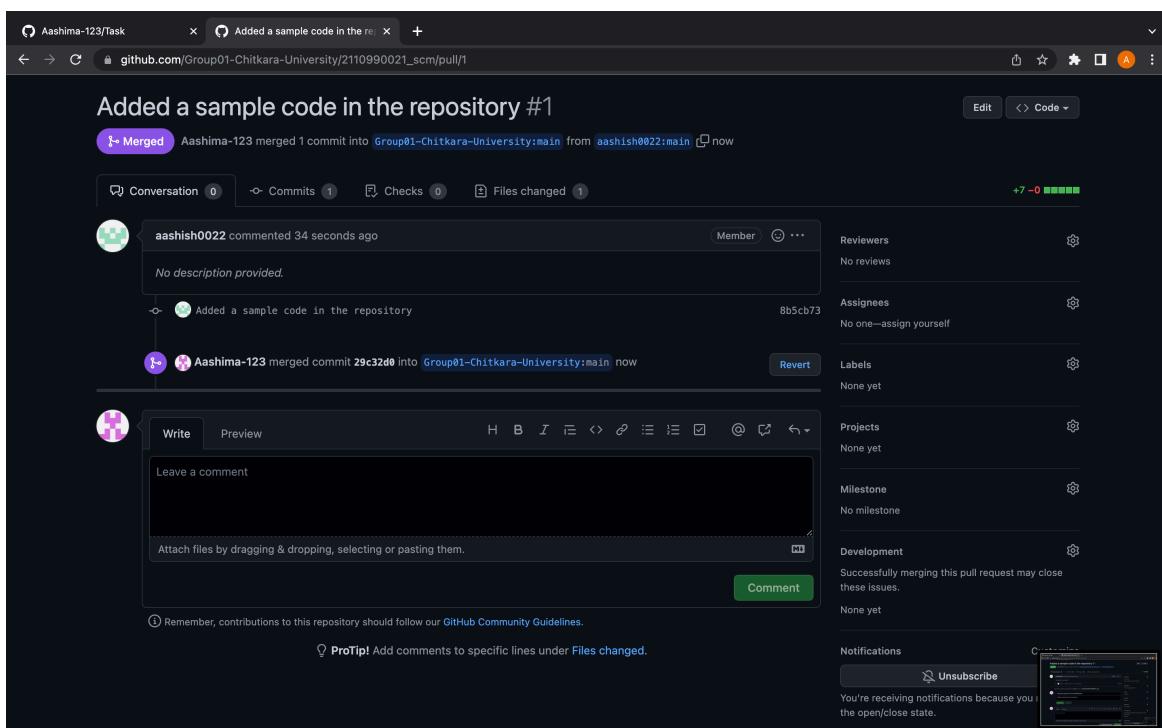
- Do the required changes in the repository, add and commit these changes in the local repository in a new branch.
- Push the modified branch using `git push origin branchname`.
- Open a pull request by following the procedure from the above experiment.
- The pull request will be created and will be 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 the pull request. Two option will be available, either to close the pull request or Merge the request with the main branch.
- By selecting the merge branch option the main branch will get updated for all the team members.



- We can merge the pull request



Experiment No. 04

Aim: Publish and print network graphs

The network graph is one of the useful features for developers on GitHub. It is used to display the branch history of the entire repository network, including branches of the root repository and branches of forks that contain commits unique to the network.

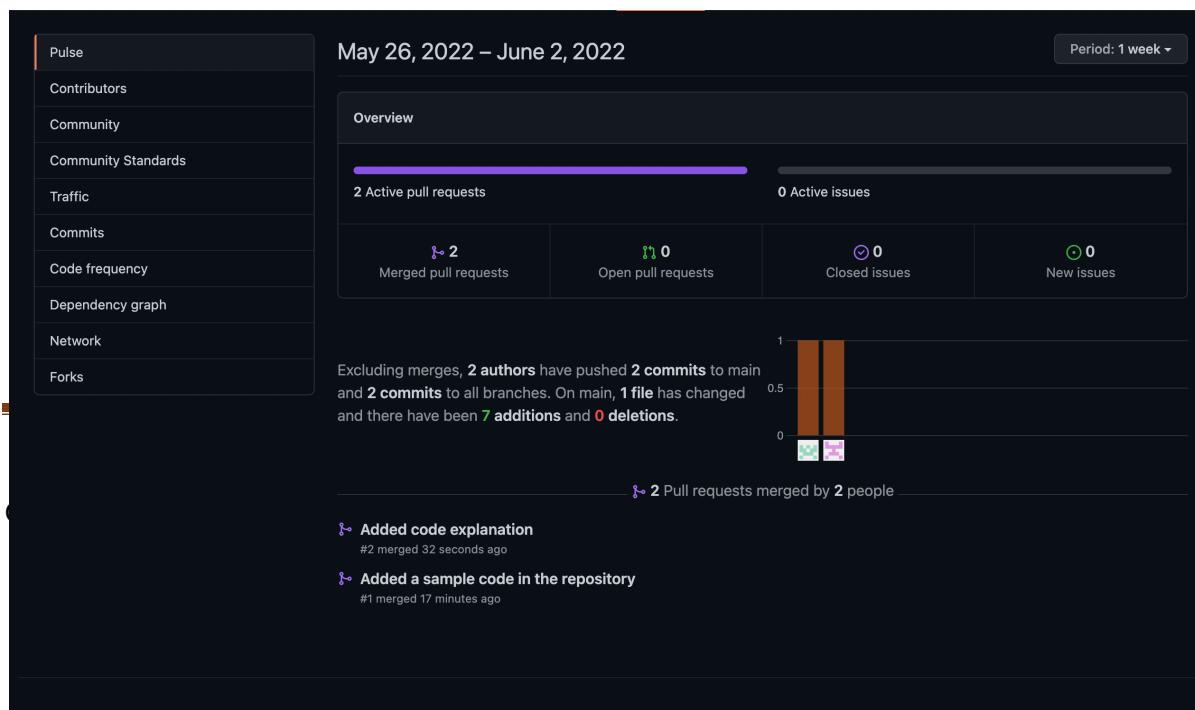
A repository's graphs give you information on traffic, projects that depend on the repository, contributors and commits to the repository, and a repository's forks and network. If you maintain a repository, you can use this data to get a better understanding of who's using your repository and why they're using it.

Some repository graphs are available only in public repositories with GitHub Free:

- Pulse
- Contributors
- Traffic
- Commits
- Code frequency
- Network

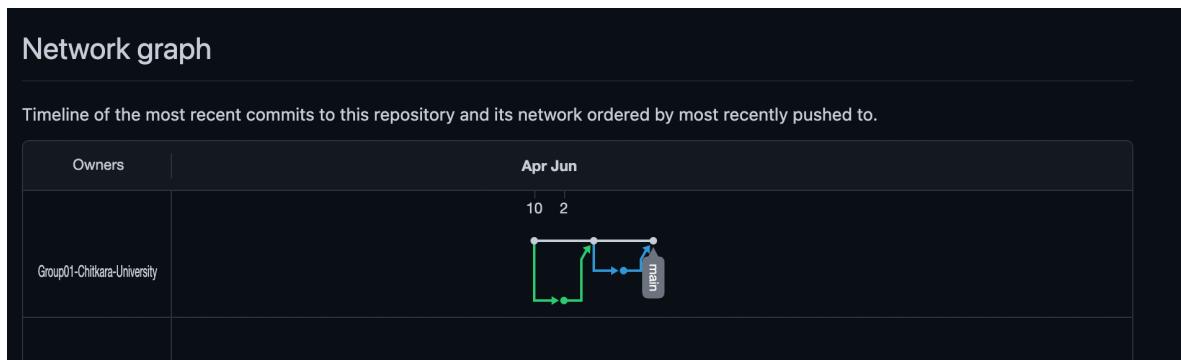
Steps to access network graphs of respective repository

1. On GitHub.com, navigate to the main page of the repository.
2. Under your repository name, click **Insights**.



3. At the left sidebar, click on **Network**.

You will get the network graph of your repository which displays the branch history of the entire repository network, including branches of the root repository and branches of forks that contain commits unique to the network.



Pull Requests of all the team members:

Team Member 2: Aashima Mahajan

1. Closed:

a. From Aashish0022

The screenshot shows a GitHub pull request page for a repository. The title of the PR is 'Added a sample code in the repository #1'. It is marked as 'Merged' with a note that 'Aashima-123 merged 1 commit into Group01-Chitkara-University:main from aashish0022:main now'. The PR has 1 commit, 0 checks, and 1 file changed. The right sidebar contains various status metrics: 7 reviews (0 pending), 0 assignees (No one—assign yourself), 0 labels (None yet), 0 projects (None yet), 0 milestones (No milestone), and 0 development tasks (None yet). A note at the bottom states 'Successfully merging this pull request may close these issues.' There is also a 'Comment' button and a note about GitHub Community Guidelines.

b. From abhi7ts

The screenshot shows a GitHub pull request page for a repository named 'Group01-Chitkara-University/2110990021_scm'. The pull request has been merged by 'Aashima-123' into the 'main' branch. The commit message is 'Added code explanation #2' and it includes a link to 'Added code explanation'. The commit was made by 'abhi7ts' and merged by 'Aashima-123'. The pull request has 1 merge commit, 1 file changed, and 0 checks. The commit message says 'No description provided.' and 'Added code explanation'. The pull request has 1 review, 1 assignee, and 1 label. It is associated with 1 project, 1 milestone, and 1 development item. Notifications are customized, and there is an option to unsubscribe.

2.Opened:

a. In abhi7ts

The screenshot shows a GitHub pull request page. At the top, it says "Added a sample code in the repository #1". Below that, there's a green button labeled "I'm Open" and a message from "Aashima-123" stating "No description provided." and "Added a sample code in the repository". A note says "This branch has no conflicts with the base branch". On the right side, there are sections for "Reviewers" (none), "Assignees" (none), "Labels" (none), "Projects" (none), "Milestone" (none), and "Development" (none). In the bottom right corner, there's a "Comment" button.

b. In Aashish0022

This screenshot is identical to the one above, showing the same GitHub pull request details and interface.