

A Project
report on

“Task 2”

with

Source Code Management

(CS181)

Submitted by

Name :-Aashish Bhardwaj Roll No. :-2110990022

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).

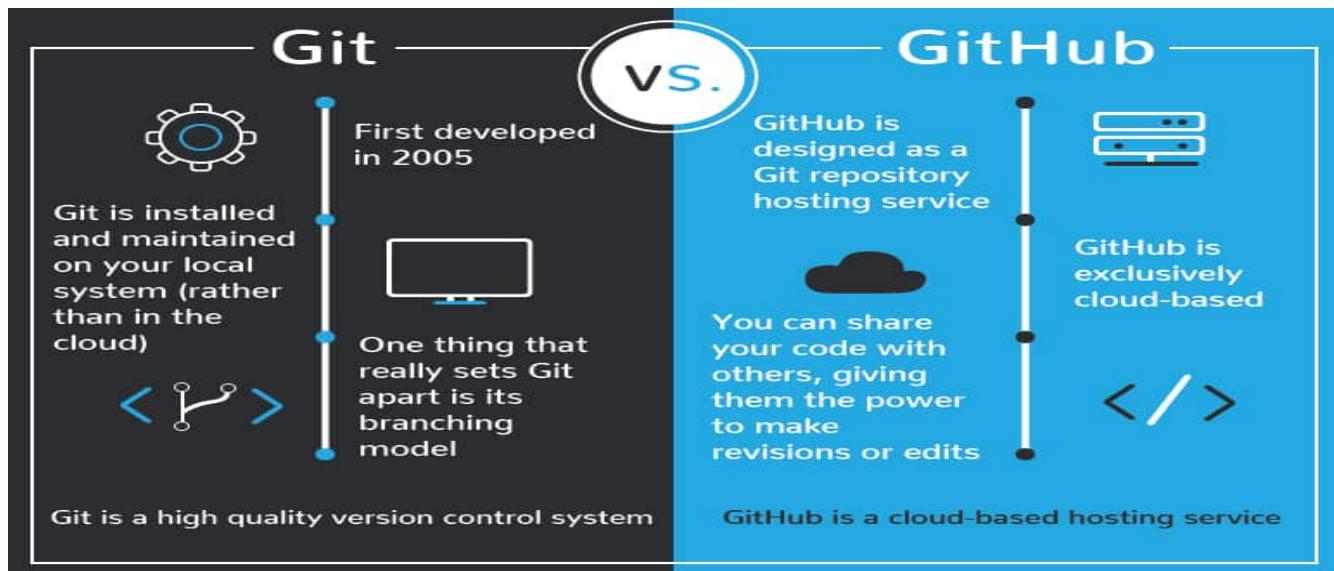


What is GITHUB?

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

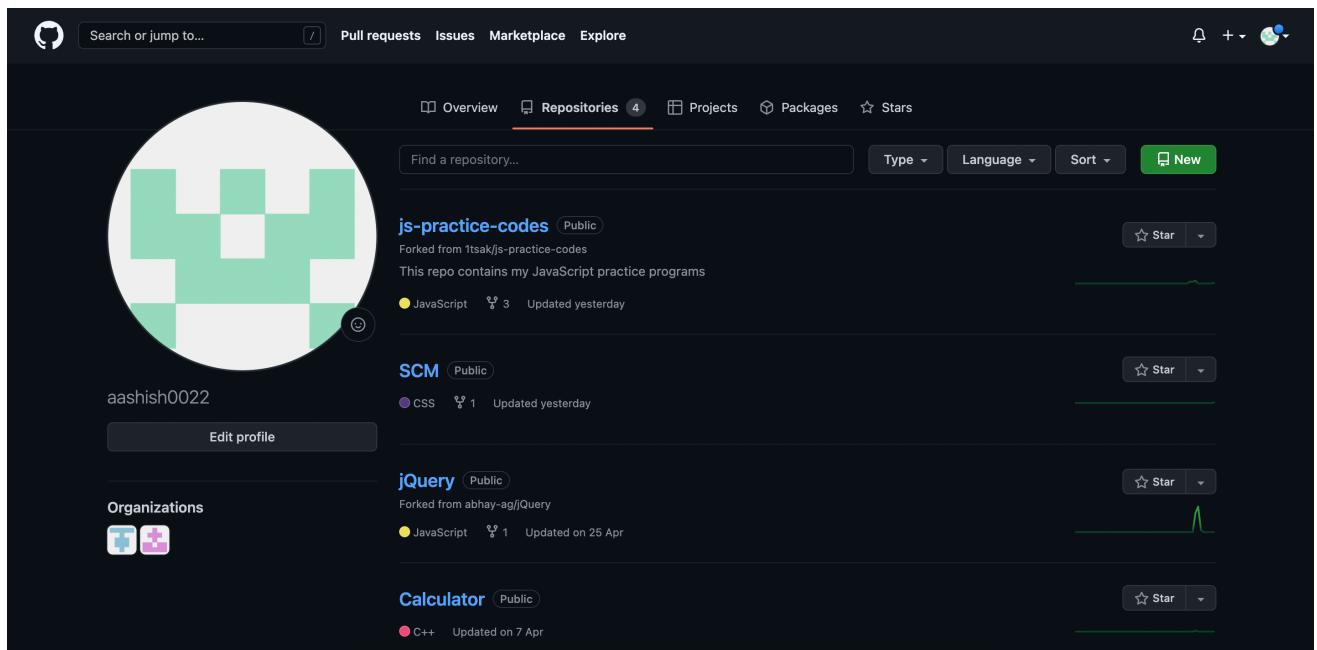
Version Control Systems is if the central server crashes, almost everything related to the project will be lost.

- 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

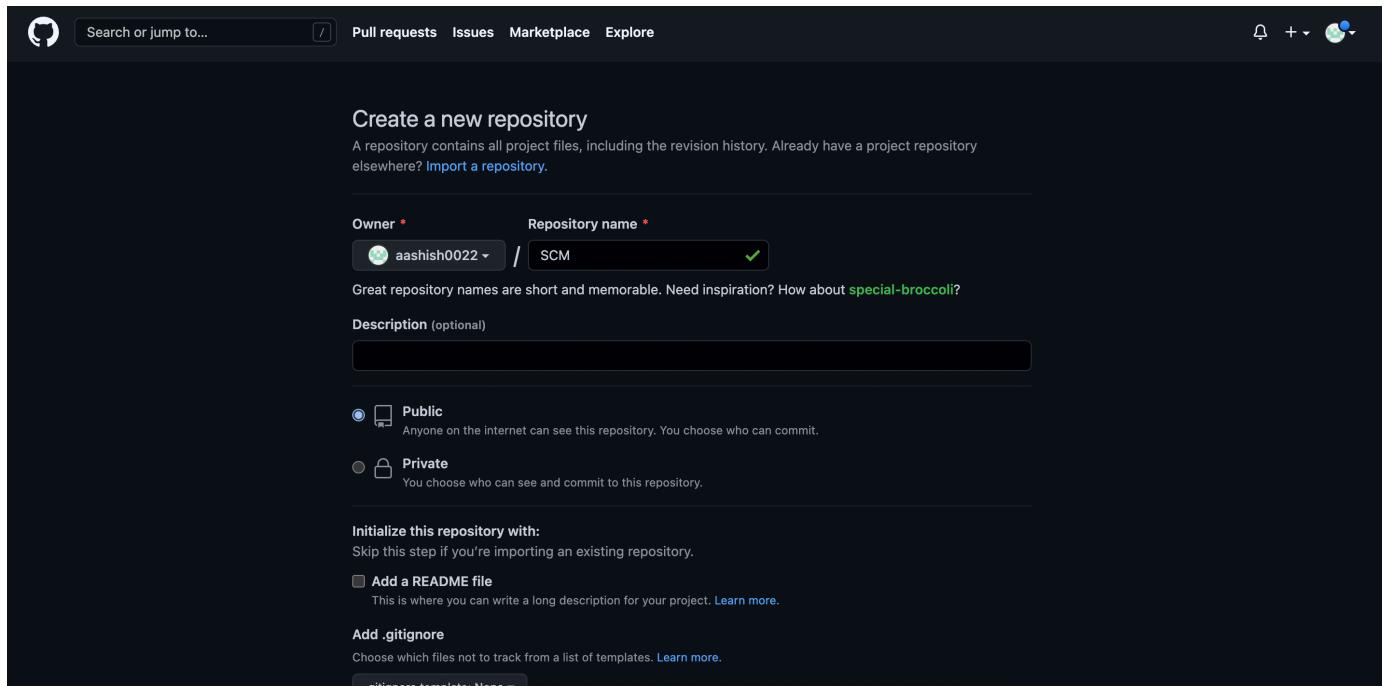
Aim: Create a distributed Repository and add members in project team

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

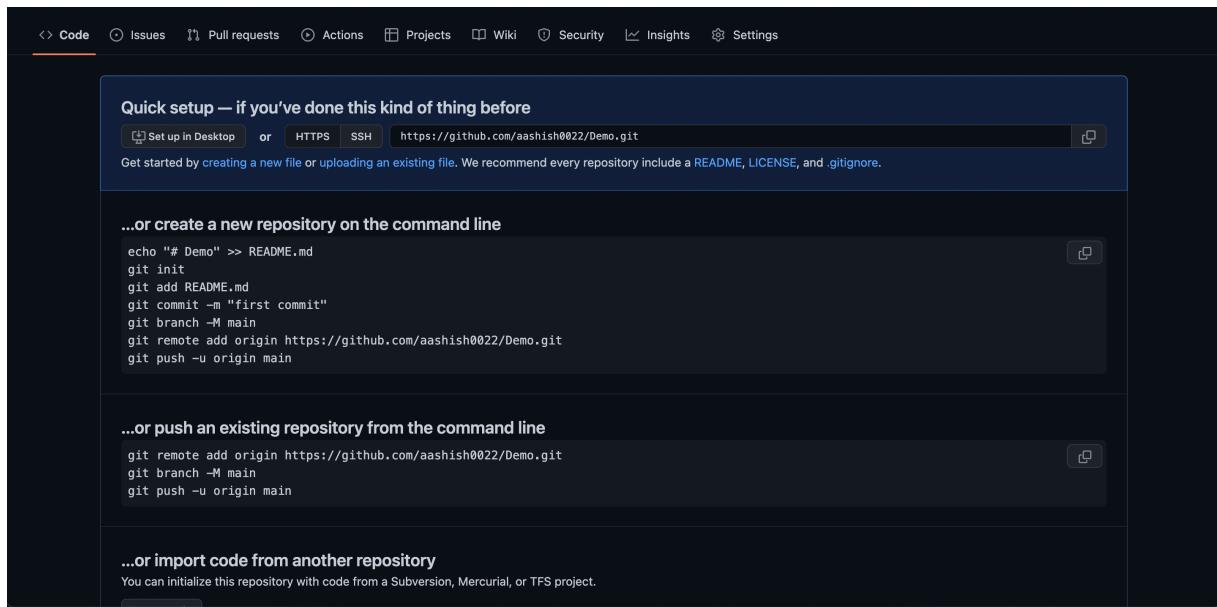


- Click on the 'New' button in the top right corner.
- Enter the Repository name and add the description of the repository.

- 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.

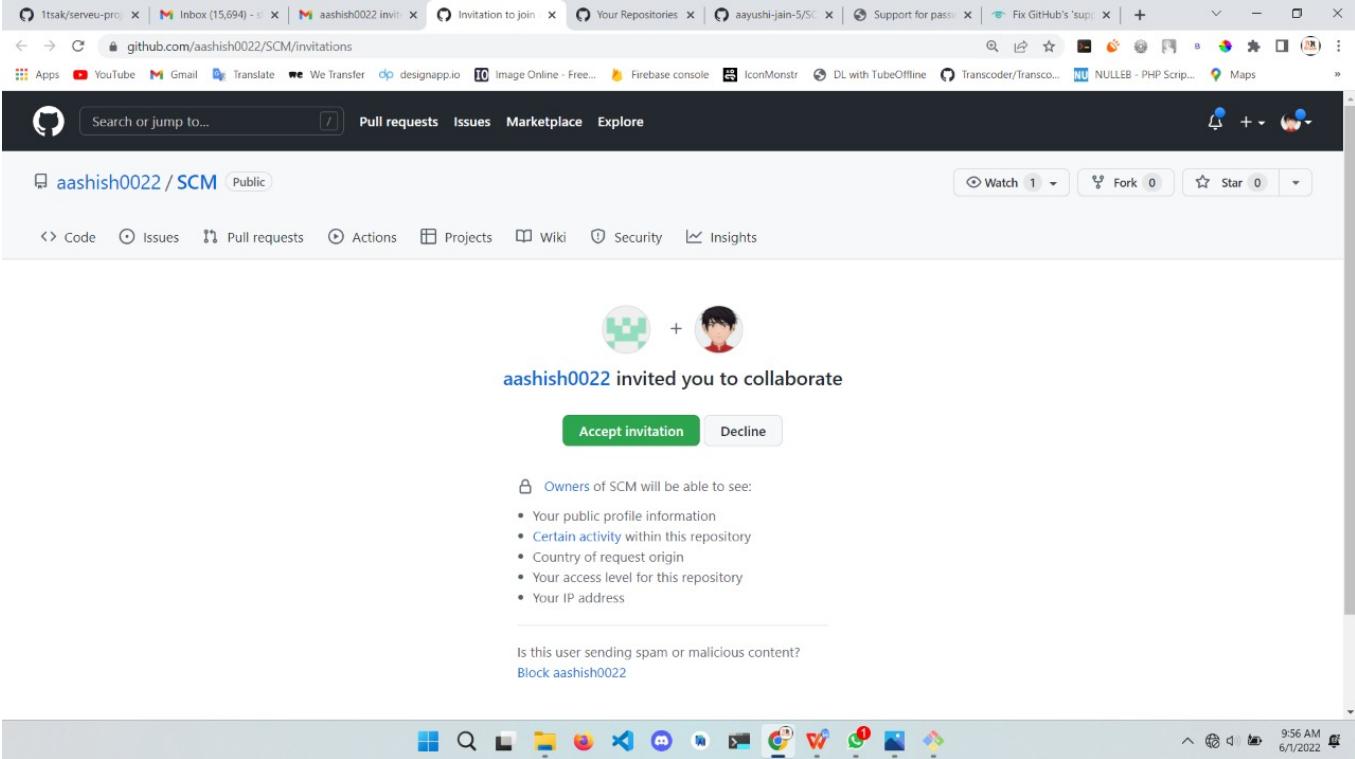


- Now, you have created your repository successfully.
- To add members to your repository open your repository and select settings option in the navigation bar.

The screenshot shows a GitHub repository named 'aashish0022/SCM' with a public status. The 'Code' tab is selected, displaying the 'master' branch. The commit history shows four commits from 'aashish0022' made yesterday, all labeled 'Initial commit'. The commits are for files '.DS_Store', 'home.html', and 'style2.css'. At the top right, there are buttons for 'Pin', 'Unwatch', 'Fork', 'Star', and more. Below the code area, there's a footer with links to GitHub's terms, privacy, security, status, docs, contact, pricing, API, training, blog, and about sections.

- 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.

The screenshot shows the 'Settings' tab of the repository. On the left, a sidebar lists various settings categories: General (selected), Access, Collaborators, Moderation options, Code and automation (Branches, Tags, Actions, Webhooks, Environments, Pages), Security, Code security and analysis, Deploy keys, and Secrets. The 'General' tab contains fields for 'Repository name' (set to 'SCM') and a 'Rename' button. It also includes a 'Template repository' checkbox and a 'Social preview' section with instructions for uploading an image. A large empty box is provided for social media preview images.



- To remove any member click on remove option available inthe last column of member's respective row.
- To accept the invitation from your team member, open youremail registered with Github.
- You will receive an invitation mail from the repositoryowner. Open the email and click on accept invitation.

The screenshot shows the 'Who has access' section of a GitHub repository settings page. On the left, there's a sidebar with sections like General, Access (Collaborators selected), Code and automation, Security, and others. The main area shows that the repository is public and has one collaborator with direct access. A 'Manage' button leads to a 'Manage access' interface where 'Akash Jha' is listed as a collaborator. There are buttons for 'Add people' and 'Remove'.

- 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.

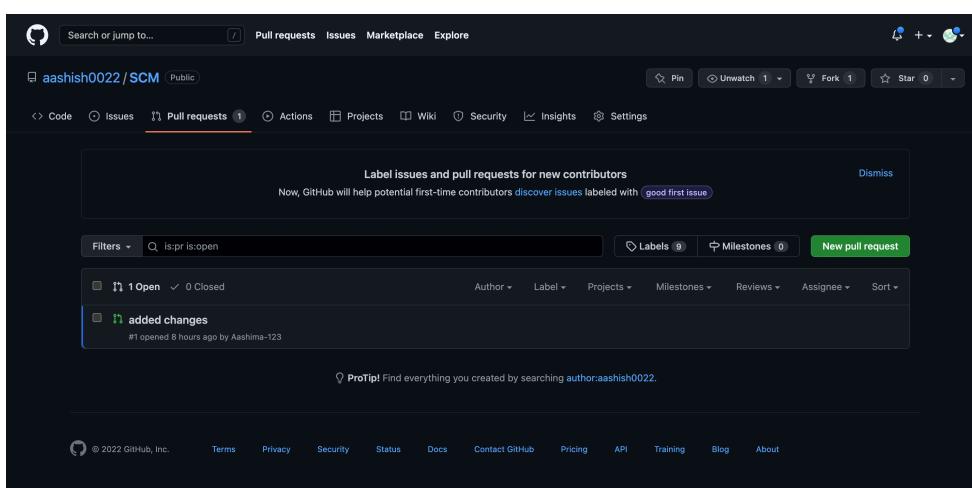
Experiment No. 02

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.

Use git push origin *branchname* option to push the new branch to themain repository.

- After pushing new branch Github will either automatically ask you tocreate a pull request or you can create your own pull request.
- To create your own pull request click on pull request option.

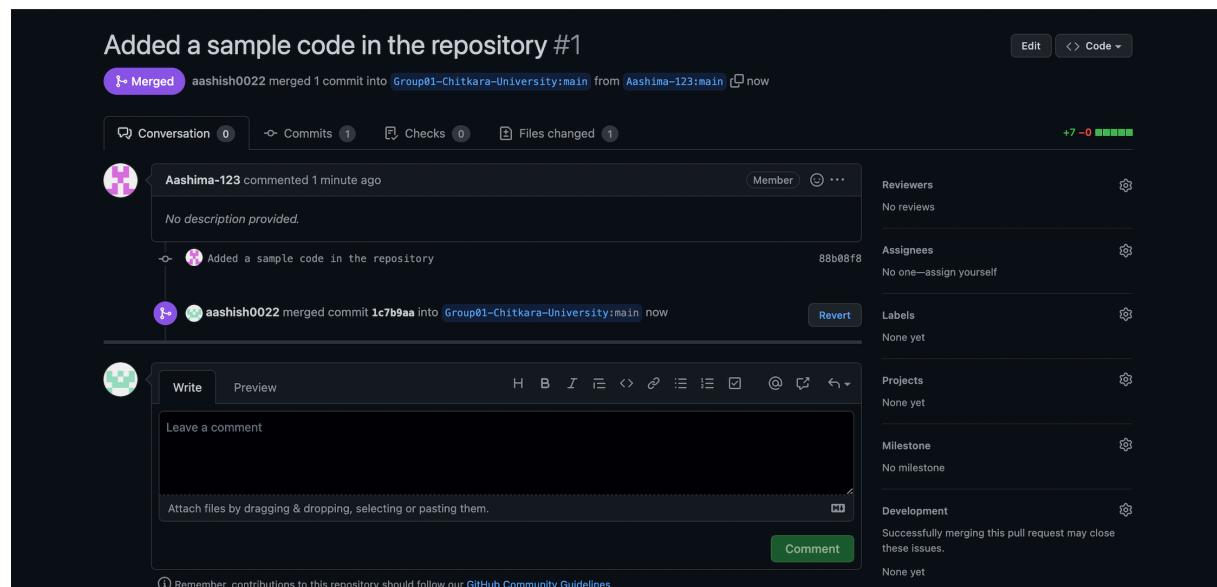


Pull Requests of all the team members:

Team Member 1: Aashish Bhardwaj

1. Closed:

a. From Aashima-123



b. From abhi7ts

A screenshot of a GitHub pull request merge commit page. The title is "Added code Explanation #2". A purple button at the top left says "Merged" with the message "aashish0022 merged 1 commit into Group01-Chitkara-University:main from abhi7ts:main now". Below this are tabs for Conversation (0), Commits (1), Checks (0), and Files changed (1). A comment from user "abhi7ts" is shown, stating "No description provided." and "Added code Explanation". A merge commit from "aashish0022" is shown with the commit hash "7911a16" and a "Revert" button. On the right, there are sections for Reviewers (No reviews), Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), Milestone (No milestone), and Development (Successfully merging this pull request may close these issues. None yet). At the bottom, there's a toolbar with icons for GitHub, Slack, and other services, and a "Comment" button.

2.Opened:

a. In abhi7ts

A screenshot of a GitHub pull request creation page. The URL is "github.com/Group01-Chitkara-University/2110990041/pull/2". The tab bar shows "Issues", "Pull requests" (1), "Actions", "Projects", "Wiki", "Security", and "Insights". The title is "Added code explanation #2". A green button at the top left says "Open" with the message "aashish0022 wants to merge 1 commit into Group01-Chitkara-University:main from aashish0022:main now". Below this are tabs for Conversation (0), Commits (1), Checks (0), and Files changed (1). A comment from user "aashish0022" is shown, stating "No description provided." and "Added code explanation". A note below says "Add more commits by pushing to the main branch on aashish0022/2110990041." A green box highlights a message: "This branch has no conflicts with the base branch. Only those with write access to this repository can merge pull requests." On the right, there are sections for Reviewers (No reviews, Still in progress? Convert to draft), Assignees (No one assigned), Labels (None yet), Projects (None yet), Milestone (No milestone), and Development (Successfully merging this pull request may close this issue. None yet). At the bottom, there's a toolbar with icons for GitHub, Slack, and other services, and a "Comment" button.

b. In Aashima-123

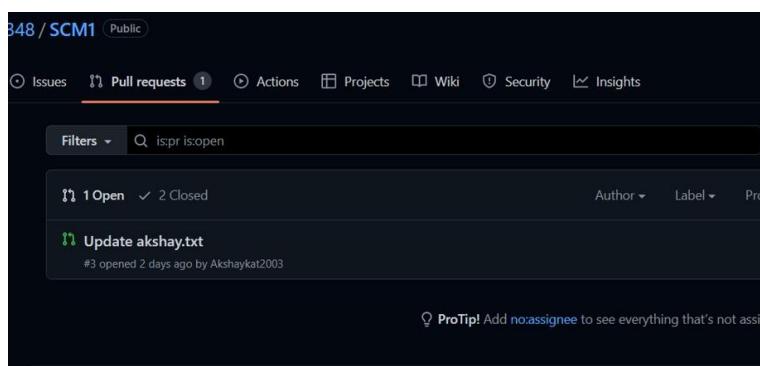
The screenshot shows a GitHub pull request page. The title of the pull request is "Added a sample code in the repository #1". The URL in the address bar is `github.com/Group01-Chitkara-University/2110990021_scm/pull/1`. The pull request has been merged, indicated by the green checkmark icon and the text "Merge pull request #1 from aashish0022:main". The commit message is "Added a sample code in the repository". The commit hash is 8b5cb73. The pull request has 0 reviews and 0 assignees. The status bar at the bottom shows 2,784 notifications.

Experiment No. 03

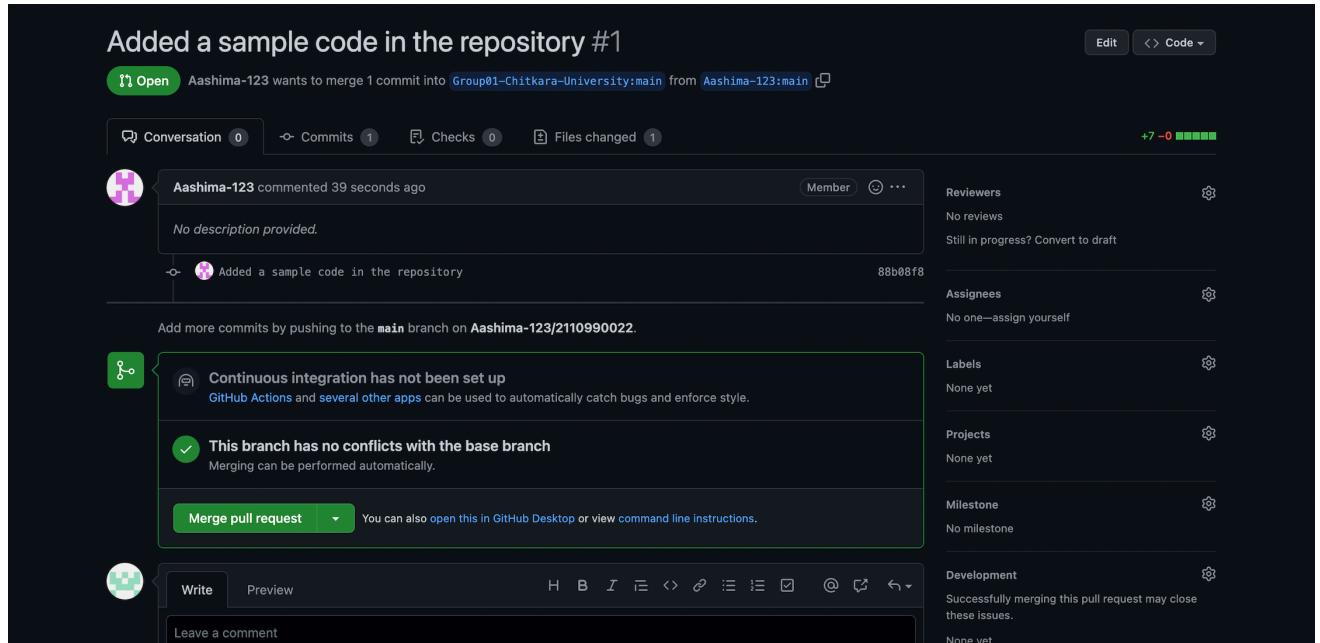
Aim: Create a pull request on a team member's repo and close pull requests generated by team members on own Repo as a maintainer

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



- 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.



Added a sample code in the repository #1

Aashima-123 wants to merge 1 commit into [Group01-Chitkara-University:main](#) from [Aashima-123:main](#)

Conversation 0 Commits 1 Checks 0 Files changed 1 +7 -0

Aashima-123 commented 39 seconds ago
No description provided.

Added a sample code in the repository 88b08f8

Continuous integration has not been set up GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Write Preview

Leave a comment

Reviewers No reviews Still in progress? Convert to draft

Assignees No one—assign yourself

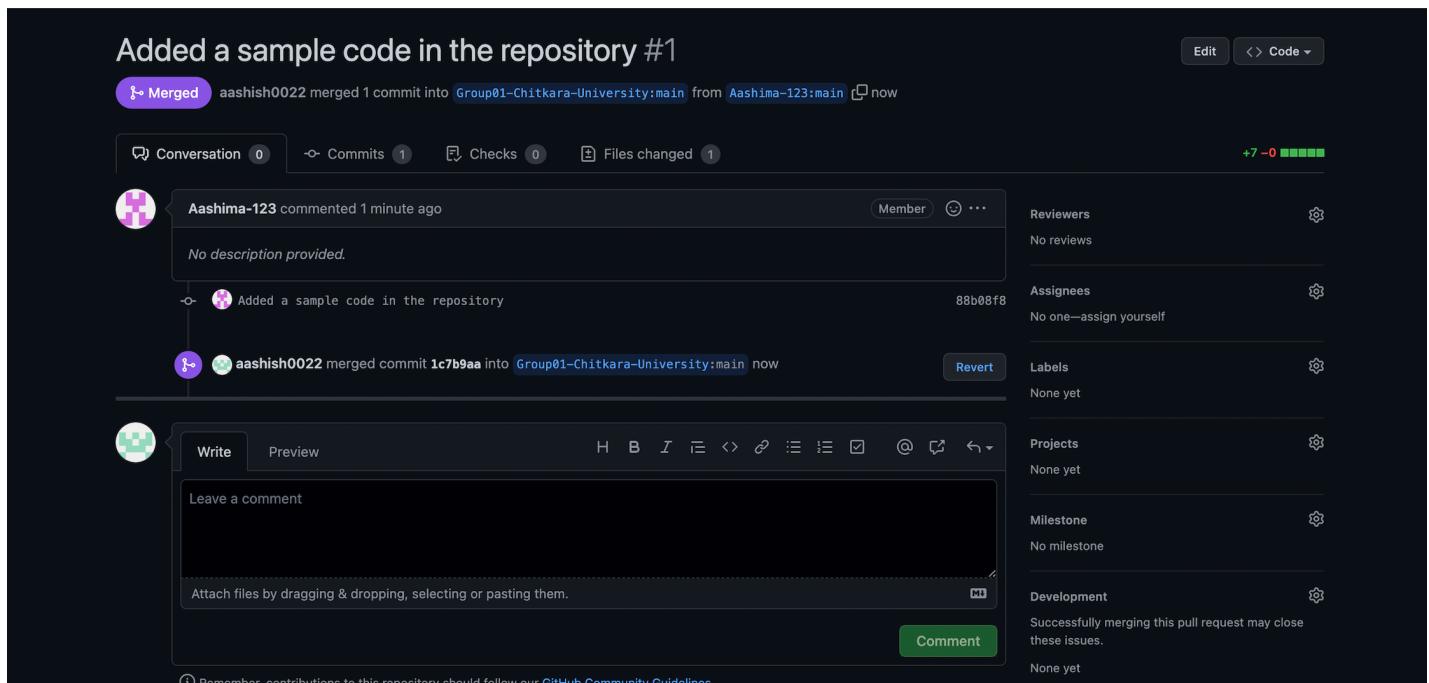
Labels None yet

Projects None yet

Milestone No milestone

Development Successfully merging this pull request may close these issues. None yet

- We can merge the pull request



Added a sample code in the repository #1

Merged aashish0022 merged 1 commit into [Group01-Chitkara-University:main](#) from [Aashima-123:main](#) now

Conversation 0 Commits 1 Checks 0 Files changed 1 +7 -0

Aashima-123 commented 1 minute ago
No description provided.

Added a sample code in the repository 88b08f8

aashish0022 merged commit [1c7b9aa](#) into [Group01-Chitkara-University:main](#) now Revert

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Comment

Reviewers 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

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.

