

TASK 2

SOURCE CODE MANAGEMENT

(CS-181)

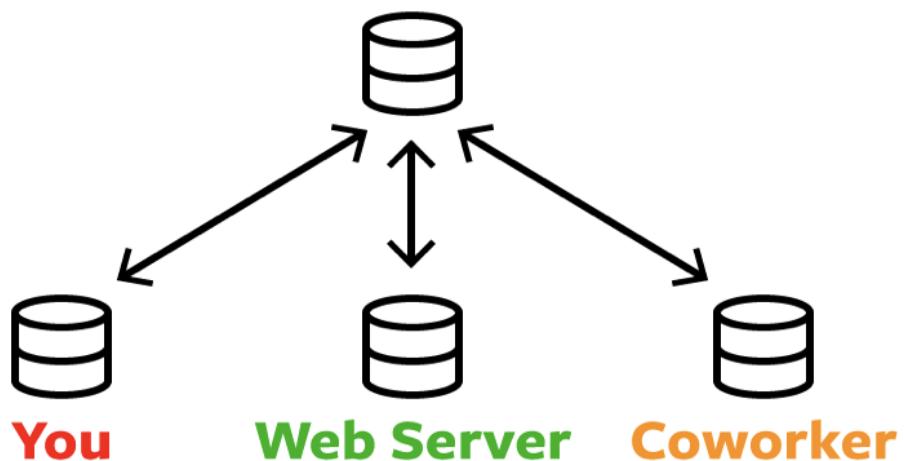
Submitted By:
Abhijit Singh
Roll no.-2110990041

Submitted To:
Dr. Monit Kapoor

WHAT IS GIT?

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. A version-controlled system is the one which keeps the history of the versions. 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.

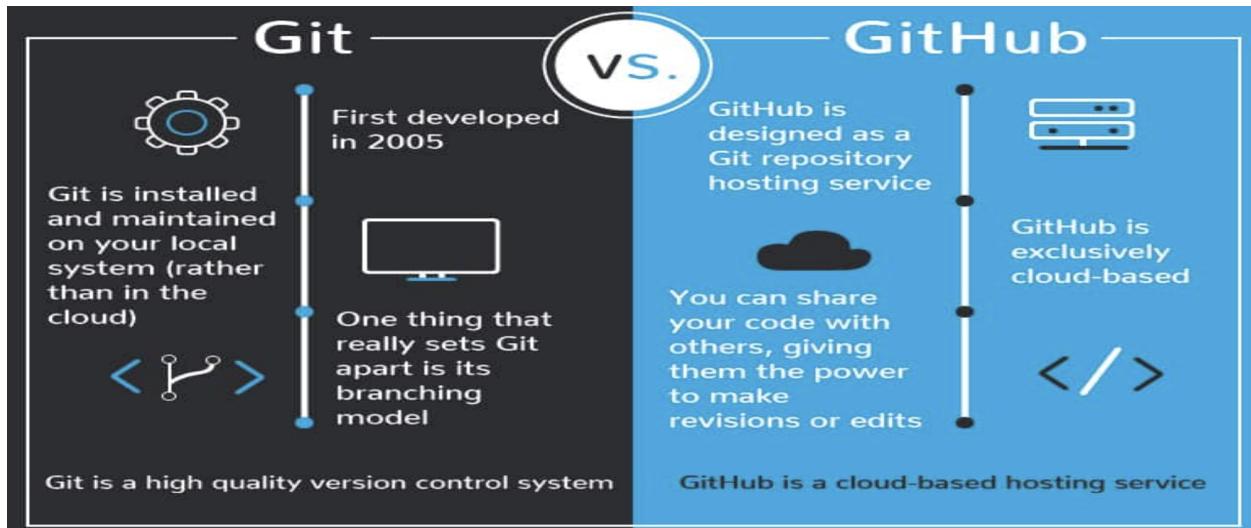
GitHub, Bitbucket, etc.



NEED OF GIT:

- Software always evolves with time and there are always different versions of the software. Keeping every version/copy of the software is not an optimized method.
- Any changes made to the source code were unknown to the other developer.
- There arose the need of software which helps in the code management and the git was developed.

What is the difference between GIT and GITHUB?



What is a 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.

EVOLUTION OF THE VERSION CONTROLLED SYSTEM:

- **LOCAL VCS:**

The 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.

- **CENTRALIZED VCS:**

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.

- **DISTRIBUTED VCS:**

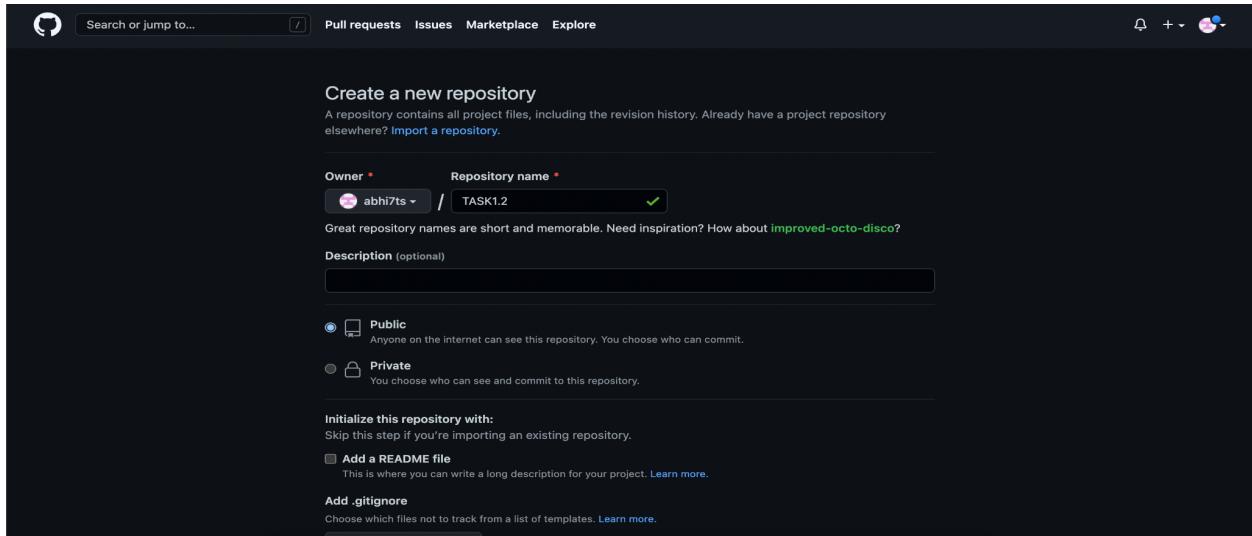
In a distributed version control system, there will be one or more servers and many collaborators similar to the centralized system. But the difference is, not only do they check out the latest version, but each collaborator will have an exact copy of the main repository on their local machines. Each 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 1

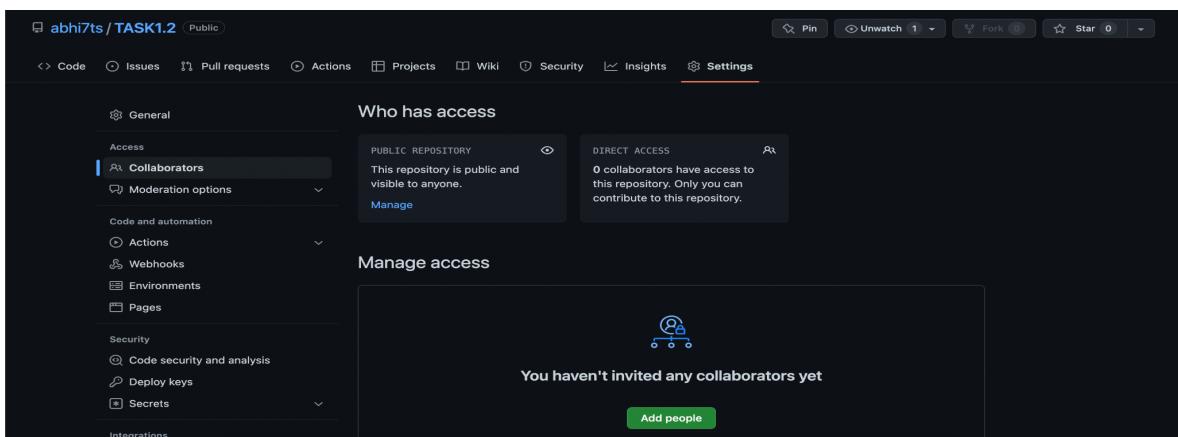
AIM: Create a distributed Repository and add members in the project team.

PROCEDURE:

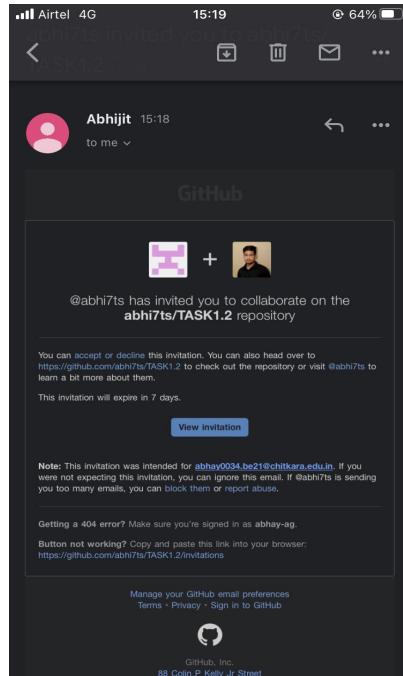
- On your GitHub account, create a new repo.



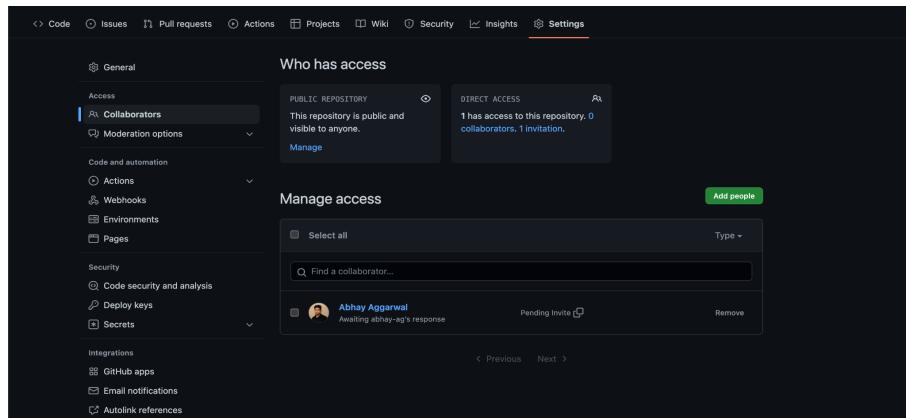
- Open repo and click on settings in the menu bar. In the access section of the sidebar, click on the collaborators and a confirm page would appear asking you to enter the password, after entering password it will show who all have the access to this



- Click on add people and a box will appear on the screen. Enter or search the username of the collaborators you want to add to the repository.
- An invitation Mail will be sent to the Collaborator which will expire after 7 days if not accepted.



- If the user accepts the invitation, then the collaborators will have access to this repository.

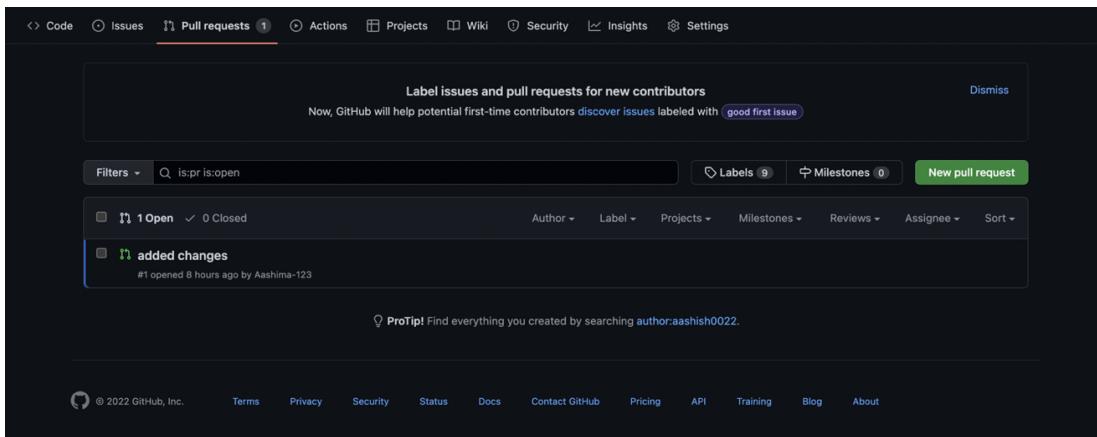


Experiment 2

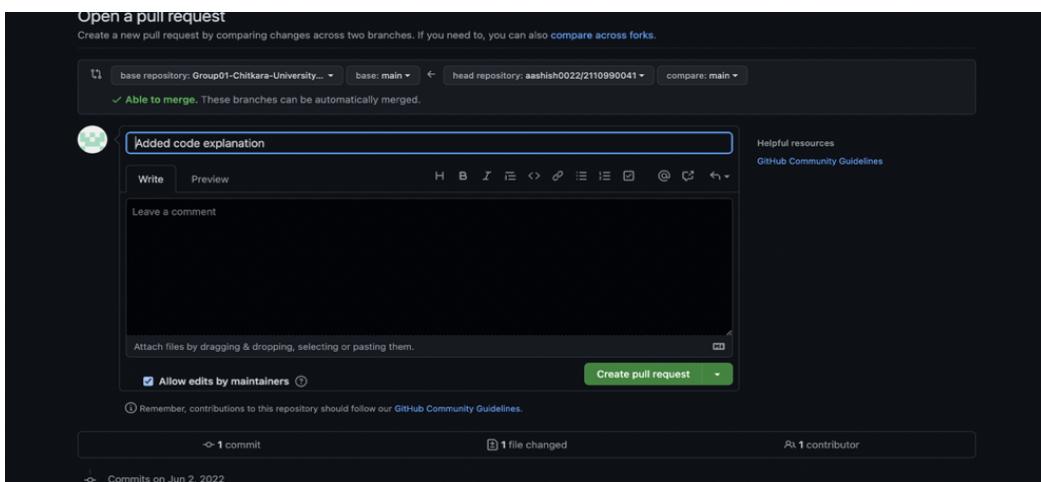
AIM: Open And Close a Pull Request.

PROCEDURE:

- To open a pull request we first have to make a new branch, by using the git branch branchname option.
- After making a 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 the git push origin branchname option to push the new branch to the main repository. After pushing a 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.



- 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 your 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 requests generated and how they were dealt with by clicking on pull request option.

The screenshot shows a GitHub interface with the 'Pull requests' tab selected. A search bar at the top contains the query 'is:pr is:closed'. Below the search bar, there are filter options: '0 Open' and '2 Closed'. The 'Closed' section lists two pull requests:

- #2 by abhinav-lyagid3 added new file again (closed 44 minutes ago)
- #1 by abhinav-lyagid3 New file added (merged 1 hour ago)

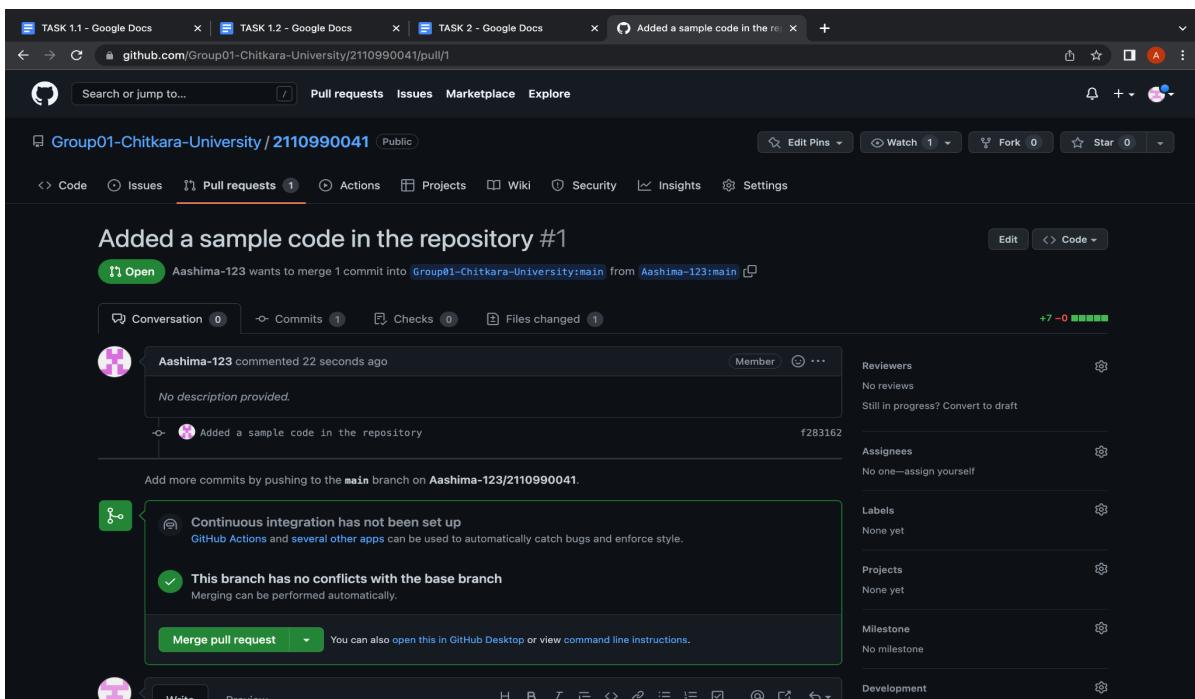
A red circle highlights the second pull request, #2. At the bottom of the page, there is a footer with links to GitHub's Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About pages.

Experiment 3

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

PROCEDURE:

- 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 branch option the main branch will get updated for all the team members. We can merge the pull request

The screenshot shows a GitHub pull request page for a repository named "Group01-Chittkara-University". The pull request is titled "Added a sample code in the repository #1". A purple "Merged" button indicates that the pull request has been merged. The commit message is "Added a sample code in the repository". The pull request has 7 additions and 0 deletions. The right sidebar shows various review and management details: no reviews, no assignees, no labels, no projects, no milestones, and no development notes. A comment section at the bottom allows users to leave comments or attach files.

Experiment 4

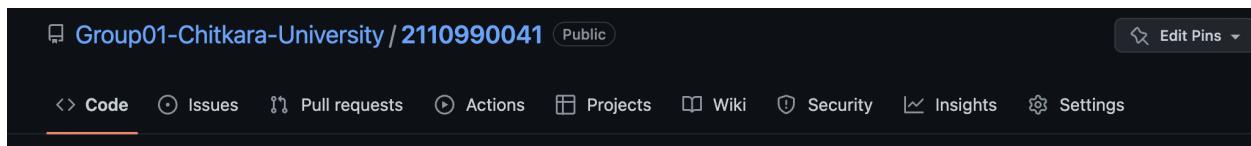
AIM: Publish and print network graphs.

THEORY: 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

PROCEDURE:

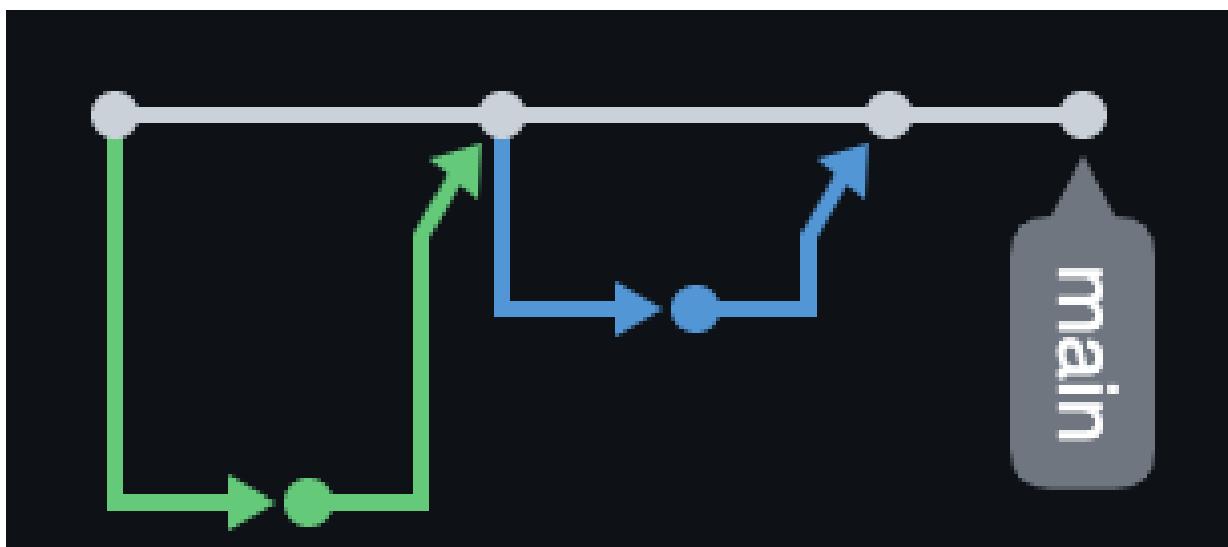
- Goto the Insights tab of the Repository.



- 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

CLOSED:

- From aashish0022

The screenshot shows a GitHub pull request page for a repository named 'Group01-Chitkara-University:main'. The pull request has been merged by 'abhi7ts' from 'aashish0022:main' into 'Group01-Chitkara-University:main'. The commit message is 'Added code explanation'. The pull request has 1 review, 1 commit, 0 checks, and 1 file changed. The conversation shows a comment from 'aashish0022' stating 'No description provided.' and a merge commit from 'abhi7ts'. The right sidebar contains sections for Reviewers, Assignees, Labels, Projects, Milestone, Development, and Notifications.

- From Aashima-123

The screenshot shows a GitHub pull request page for a repository named 'Group01-Chitkara-University:main'. The pull request has been merged by 'abhi7ts' from 'Aashima-123:main' into 'Group01-Chitkara-University:main'. The commit message is 'Added a sample code in the repository'. The pull request has 7 reviews, 0 commits, 0 checks, and 1 file changed. The conversation shows a comment from 'Aashima-123' stating 'No description provided.' and a merge commit from 'abhi7ts'. The right sidebar contains sections for Reviewers, Assignees, Labels, Projects, Milestone, Development, and Notifications.

OPENED:

- In Aashima-123

Group01-Chitkara-University / 2110990041 (Public)

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

Added a sample code in the repository #1

Open 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

Aashima-123 commented 22 seconds ago
No description provided.

Added a sample code in the repository f283162

Add more commits by pushing to the main branch on Aashima-123/2110990041.

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.

Reviewers No reviews Still in progress? Convert to draft

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

Development

- In aashish0022

Group01-Chitkara-University / 2110990041 (Public)

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

Added code explanation #2

Open aashish0022 wants to merge 1 commit into Group01-Chitkara-University:main from aashish0022:main

Conversation 0 Commits 1 Checks 0 Files changed 1

aashish0022 commented 1 minute ago
No description provided.

Added code explanation 9ddff13

Add more commits by pushing to the main branch on aashish0022/2110990041.

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.

Reviewers Suggestions Aashima-123 Request 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.

Notifications Customize