### Subject Name: Source Code Management

Subject Code: **22CS003**

Session: **2023-2024**

Department: **DCSE**

****

##### Submitted To: Submitted By:

##### Dr. Himanshi Babbar Komal Pahuja

##### Professor 2310992057

##### Chitkara University, Punjab Sem-II (2023-27)

**Task 2 Submission** (Week 10)

1. Create a distributed Repository and add members in project team

2. Open and close a pull request.

3. Each project member shall create a pull request on a team members repo and close pull requests generated by team members on own Repo as a maintainer.

4. Publish and print network graphs

Team Member 1 Name: Komal Pahuja Roll No. 2310992042

Team Member 2 Name: Mehak Roll No. 2310992056

Team Member 3 Name:Diksha Vadehra Roll No. 2310992592

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Practical Name** | **Teacher Sign** |
| 1 | Create a distributed Repository and add members in project team |  |
| 2 | Open and close a pull request |  |
| 3 | Create a pull request on a team members repo and close pull requests generated by team members on own Repo as a maintainer |  |
| 4 | Publish and print network graphs |  |

# EXPERIMENT 1

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

#### Procedure:

# After logging in, you will see the homepage as displayed below in your GitHub account. Select Repositories from the menu bar.

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

# 

# Add the repository's description and enter the repository name.

# Choose whether to make the repository public or private.

# 

# Choose the import code option if you wish to import code from an existing repository.

# 

# Your repository has now been successfully created.

# Open your repository and use the settings option from the navigation bar to add members. And select the Collaborators menu item from the access tab.

# 

# Open your repository and select the settings option from the navigation bar to add members. And select the Collaborators option from the access tab.

# 

# Open your GitHub email account in order to accept your team member's invitation.

# 

# The option you are now permitted to push will be displayed to you. Each person is now prepared to make a contribution to the project.

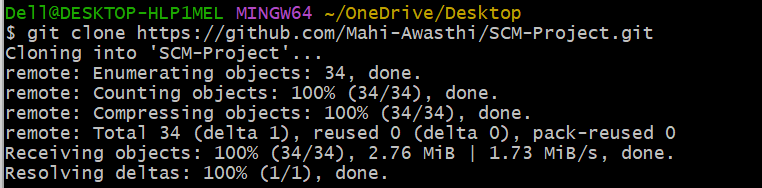
# 

# EXPERIMENT 2

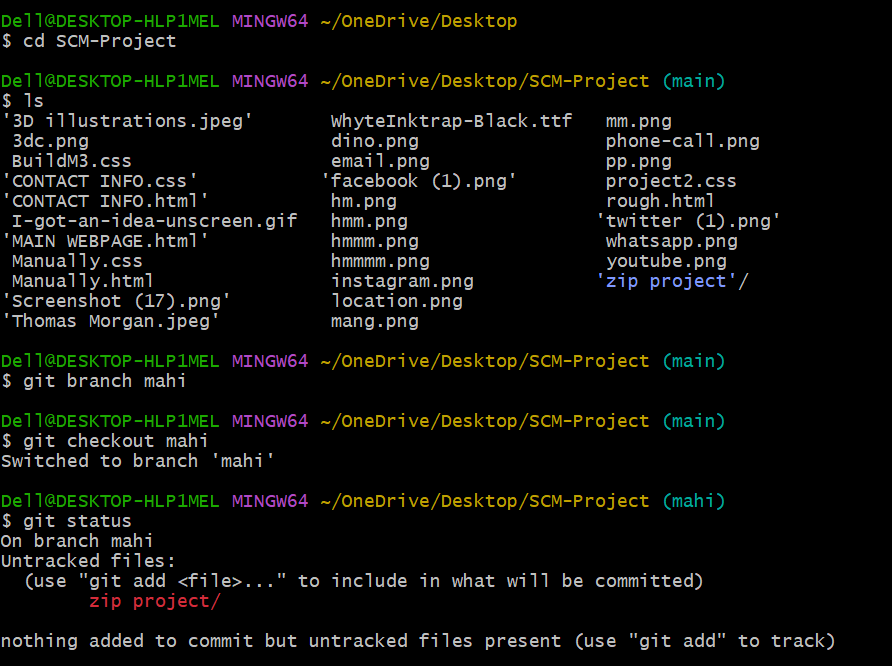
**Aim:** Open and Close a Pull Request

#### Procedure:

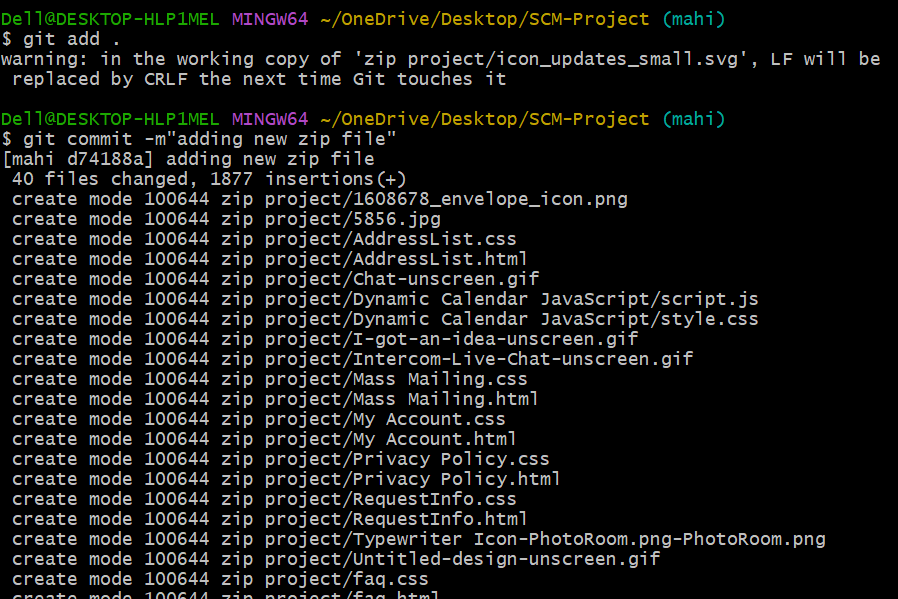
1. To initiate a pull request, first create a new branch using the git branch *branchname* option.



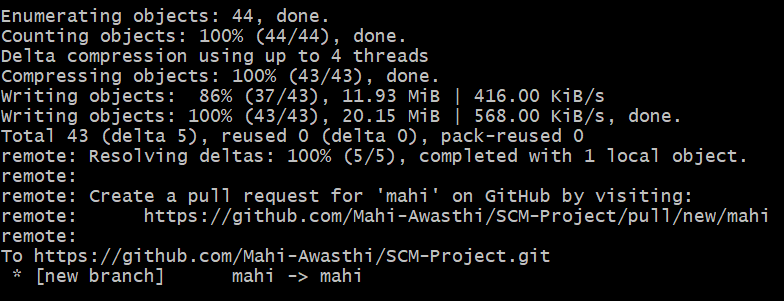
1. Once a new branch has been created, add a file to it or modify an existing file.



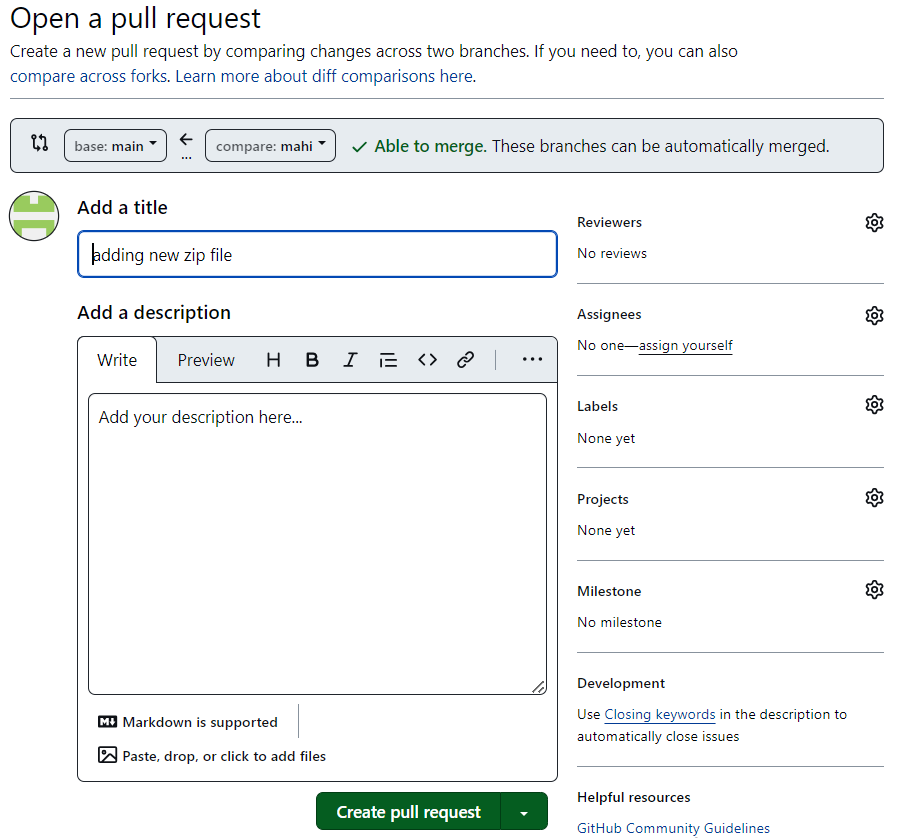
1. Commit the changes to the local repository.



1. After pushing the new branch to the main repository, use the git push origin branch name option.



1. After pushing the new branch, GitHub will either prompt you to create a pull request or you can create one on your own.
2. 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 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 request generated and how they were dealt with by clicking on pull request option.

#### 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:

#### Make the necessary repository changes, then add and commit them to a new branch in the local repository.

#### 

#### 

#### 

#### Push the modified branch using git push origin *branchname*.

#### 

#### Open a pull request by following the procedure from the above experiment.

#### Each team member will be able to see the pull request after it has been created.

#### Ask your team member to login to his/her Github account.

#### In the pull request menu, they'll see a new notification.

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

#### Select "Pull Request." There will be a choice between merging the pull request with the main branch or closing it.

#### All team members' access to the main branch will be updated upon choosing the merge branch option.

#### Closing the pull request results in it not being accepted and not being merged into the main branch.

#### The procedure is comparable to when you close and merge the pull request. All it takes is for someone else to carry it out.

#### With this, we have finished creating and closing pull requests. Additionally, we wrap up the pull request's merging into the main branch.

# EXPERIMENT 4

**Aim:** Publish and print network graphs

**Theory:**

The network graph is a valuable feature for developers on GitHub. This tool displays the repository network's branch history, including root and fork branches with network-specific commits.

Graphs for a repository include data on traffic, dependencies, contributors, commits, forks, and network activity. Maintaining a repository can provide valuable insights on user behavior and motivations.

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

• Pulse • Contributors • Traffic • Commits • Code frequency • Network

#### Procedure:

# Accessing the 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.

# 

# 4. The network graph for your repository shows the history of all branches, including the original repository and forks with unique commits.

# 

# Listing the forks of a repositoryForks are listed alphabetically by the username of the person who forked the repository

# Clicking the number of forks shows you the full network. From there you can click "members" to see who forked the repo.

# On GitHub.com, navigate to the main page of the repository.

# 

# 2. Under your repository name, click Insights.

# 3. In the left sidebar, click Forks.

# 

# 4. Here you can see all the forks.

# 

# Viewing the dependencies of a repository

# The dependency tree provides insight into the code that your repository relies on.

# Most software relies on code created and maintained by other developers, sometimes known as a supply chain. Consider utilities, libraries, and frameworks.

# Your code relies on these dependencies, so any problems or vulnerabilities can impact it.

# It is critical to assess and maintain these dependencies.