Subject Name: **Source Code Management**

Subject Code: **CS181**

Cluster: **Alpha**

Department: **CSE**

Submitted By:
YATIN DORA
2110991591
G-21

Submitted To:
DR. SHIKHA

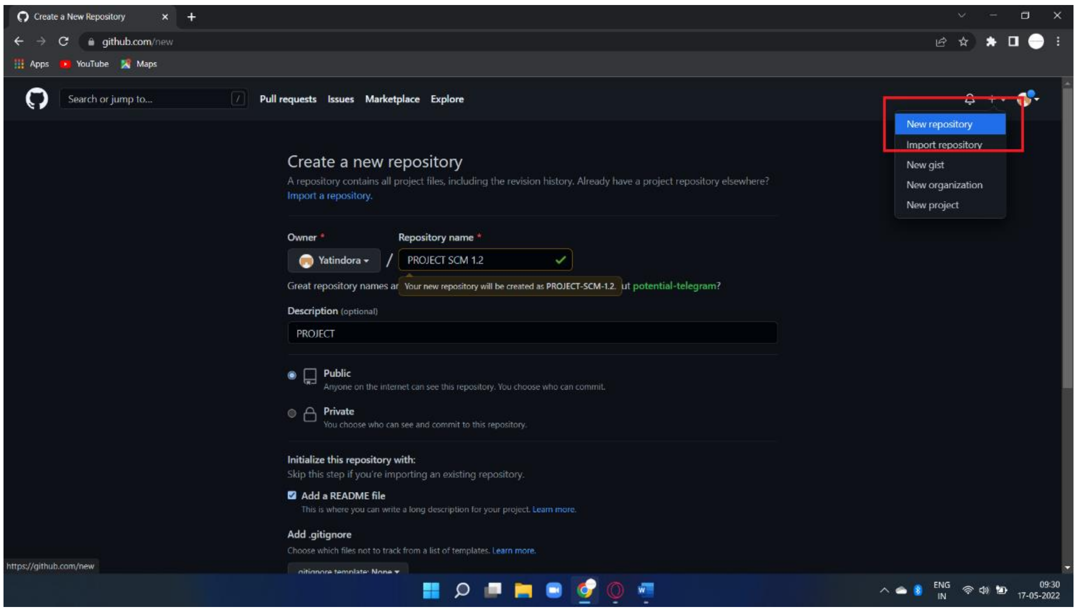
CHITKARA
UNIVERSITY

Source Code Management (Task 1.2)

EXPERIMENT- 1

AIM : Add collaborators on GitHub Repository

1. Create a New Repository.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Yatindora / Repository name * PROJECT SCM 1.2

Great repository names ar Your new repository will be created as PROJECT-SCM-1.2. ut potential-telegram?

Description (optional) PROJECT

● Public — Anyone on the internet can see this repository. You choose who can commit.

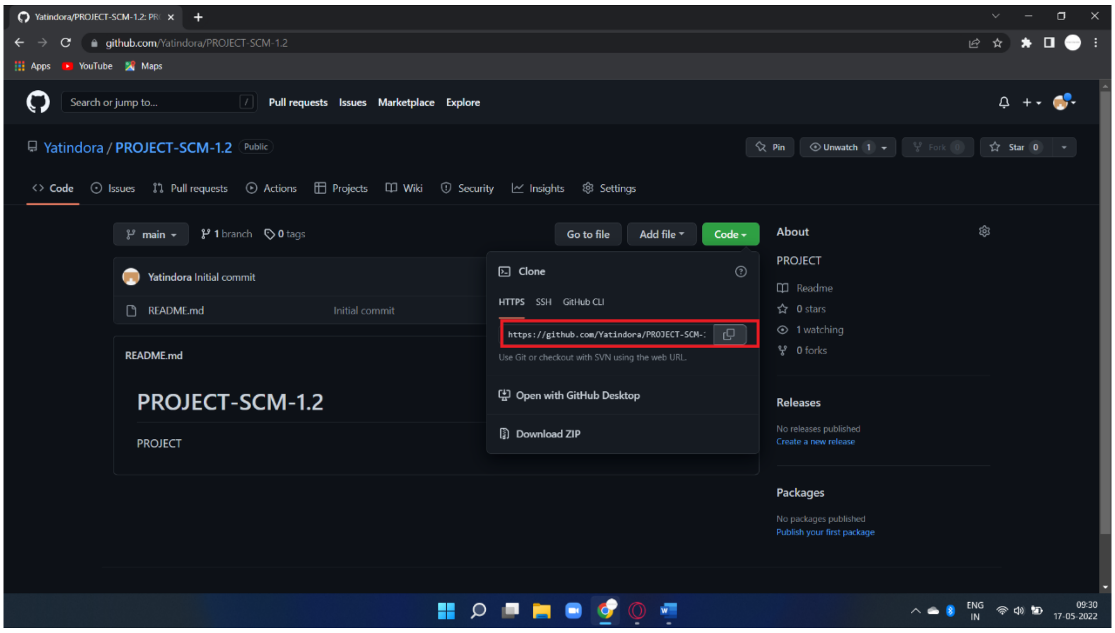
○ Private — You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☑ Add a README file — This is where you can write a long description for your project. Learn more.

Add .gitignore — Choose which files not to track from a list of templates. Learn more.

2. Now Copy the HTTP link of your repo and paste it it on your 'Git CLI', and merge the local repo in remote repository.



Yatindora Initial commit

README.md Initial commit

https://github.com/Yatindora/PROJECT-SCM-

Use Git or checkout with SVN using the web URL.

README.md

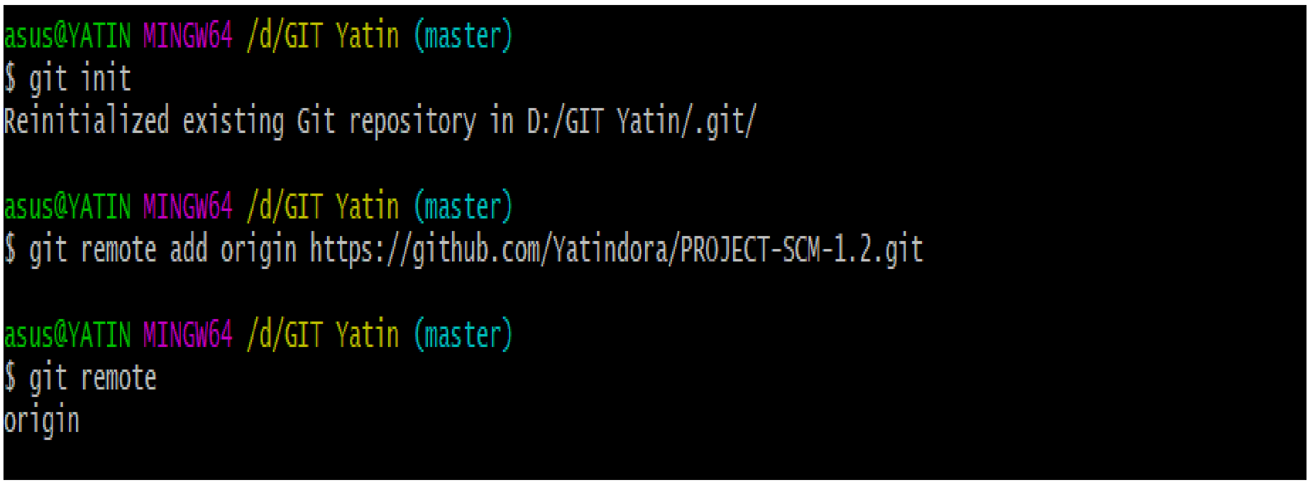
PROJECT-SCM-1.2

PROJECT

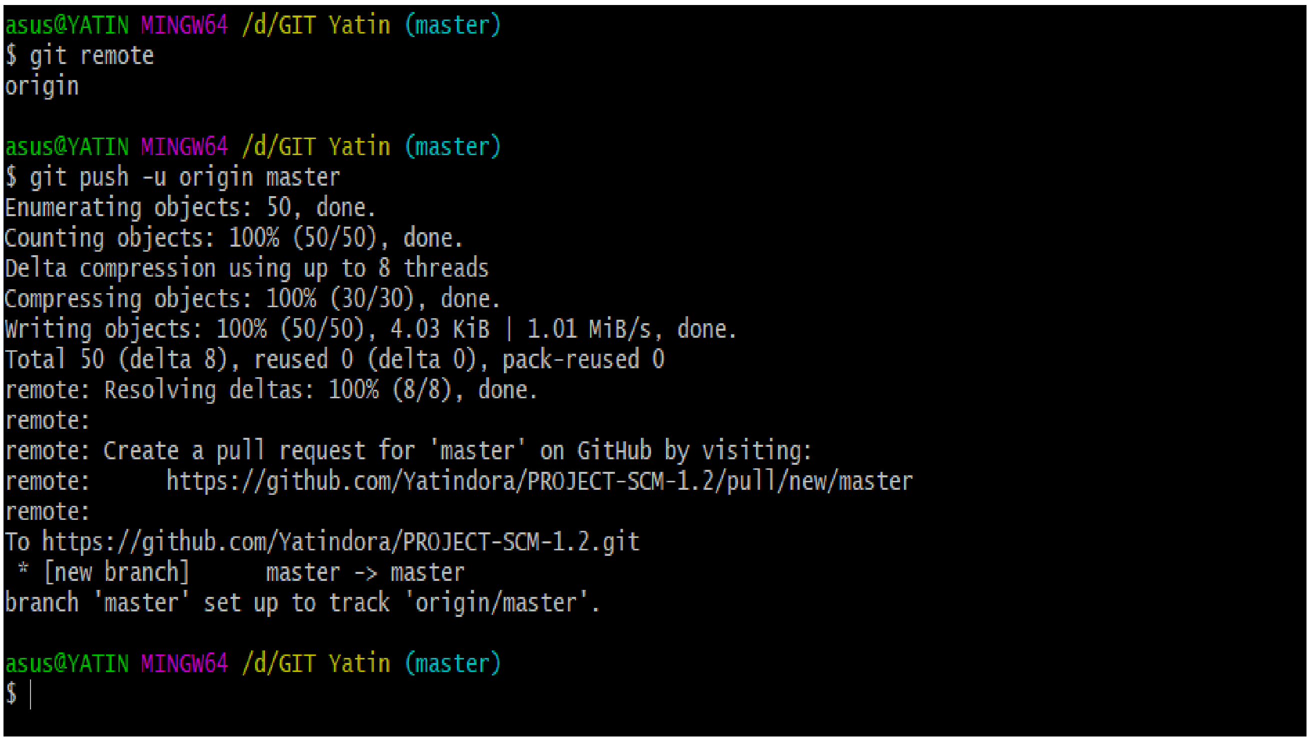
About: PROJECT — Readme, 0 stars, 1 watching, 0 forks

Releases: No releases published. Create a new release

Packages: No packages published. Publish your first package

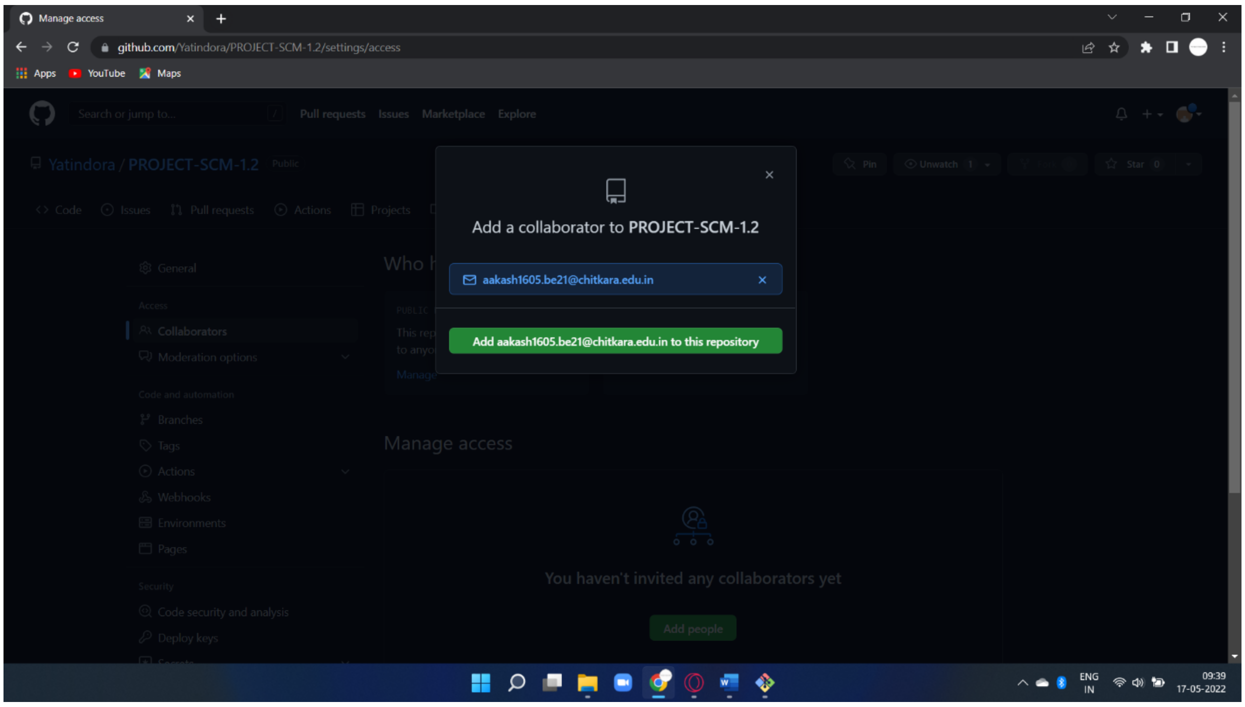


```
asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git init
Reinitialized existing Git repository in D:/GIT Yatin/.git/

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git remote add origin https://github.com/Yatindora/PROJECT-SCM-1.2.git

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git remote
origin
```



```
asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git remote
origin

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git push -u origin master
Enumerating objects: 50, done.
Counting objects: 100% (50/50), done.
Delta compression using up to 8 threads
Compressing objects: 100% (30/30), done.
Writing objects: 100% (50/50), 4.03 KiB | 1.01 MiB/s, done.
Total 50 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/Yatindora/PROJECT-SCM-1.2/pull/new/master
remote:
To https://github.com/Yatindora/PROJECT-SCM-1.2.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ |
```

3. Go to Collaborators in Repo Setting , Add the username or email of Collaborator you want to add in your Repo.



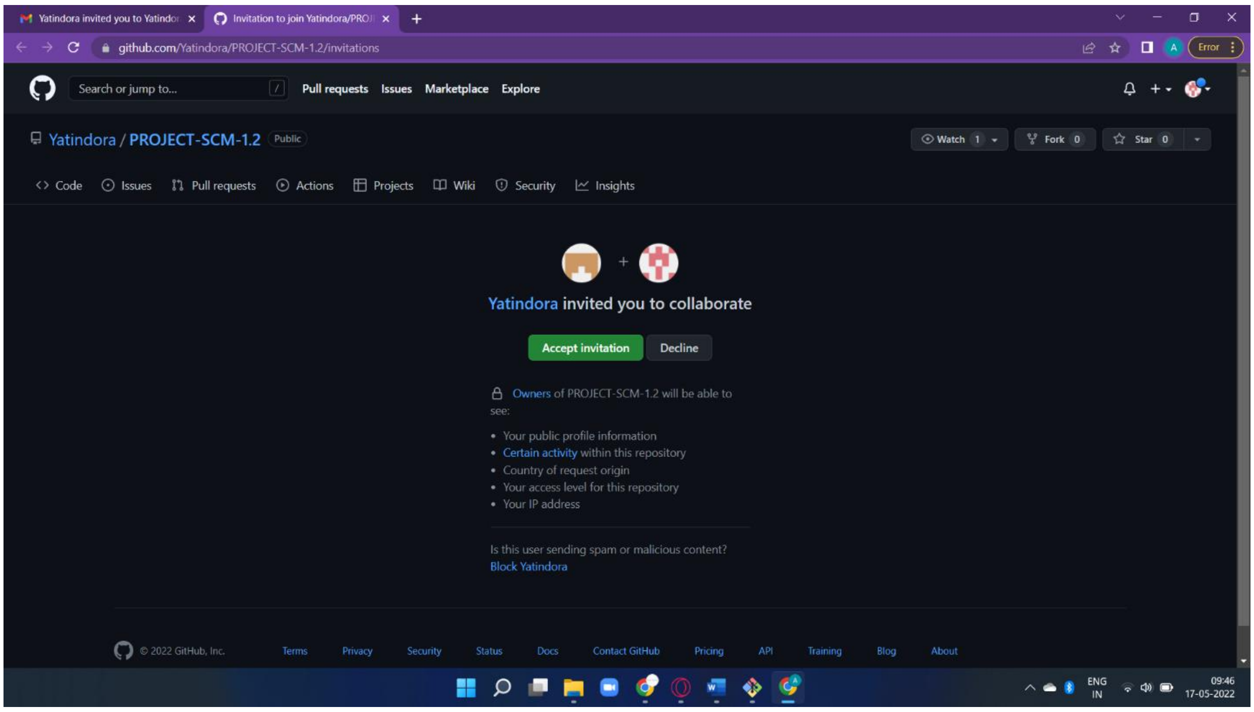
Add a collaborator to PROJECT-SCM-1.2

aakash1605.be21@chitkara.edu.in

Add aakash1605.be21@chitkara.edu.in to this repository

You haven't invited any collaborators yet

4. Invitation Mail is sent to the Collaborator, The collaborator has to accept this Invitation.



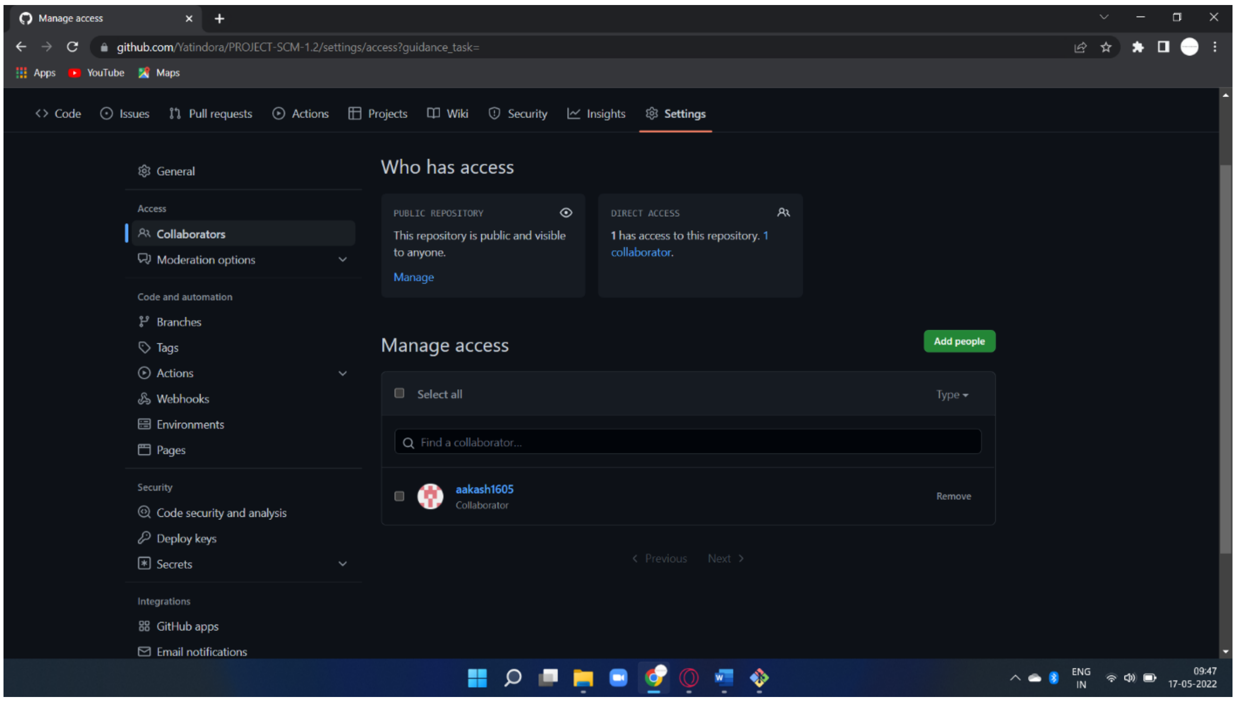
Yatindora invited you to collaborate

Owners of PROJECT-SCM-1.2 will be able to see:

- Your public profile information
- Certain activity within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?
Block Yatindora

5. New Collaborator has now access to the PROJECT SCM 1.2 .



Manage access

github.com/Yatindora/PROJECT-SCM-1.2/settings/access?guidance_task=

Apps YouTube Maps

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access
Collaborators
Moderation options

Code and automation
Branches
Tags
Actions
Webhooks
Environments
Pages

Security
Code security and analysis
Deploy keys
Secrets

Integrations
GitHub apps
Email notifications

Who has access

PUBLIC REPOSITORY
This repository is public and visible to anyone.
Manage

DIRECT ACCESS
1 has access to this repository. 1 collaborator.

Manage access

Add people

Select all

Type

Find a collaborator...

aakash1605
Collaborator

Remove

Previous Next

ENG IN 09:47 17-05-2022

EXPERIMENT- 2

Aim: FORK AND COMMIT

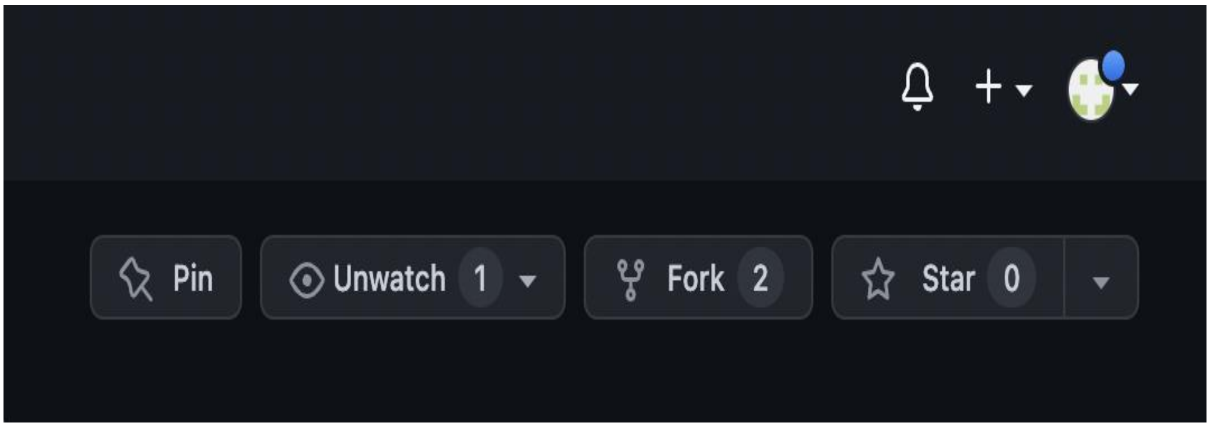
2.1 What is Forking a Repository mean and Why it is used?

Forking a repository means creating a copy of the repo. When you fork a repo, you create your own copy of the repository on your GitHub account. This is done for the following reasons:

1. You have your own copy of the project on which you may test your own changes without changing the original project.
2. This helps the maintainer of the project to better check the changes you made to the project and has the power to either accept, reject or suggest something.
3. When you clone an Open Source project, which isn't yours, you don't have the right to push code directly into the project.

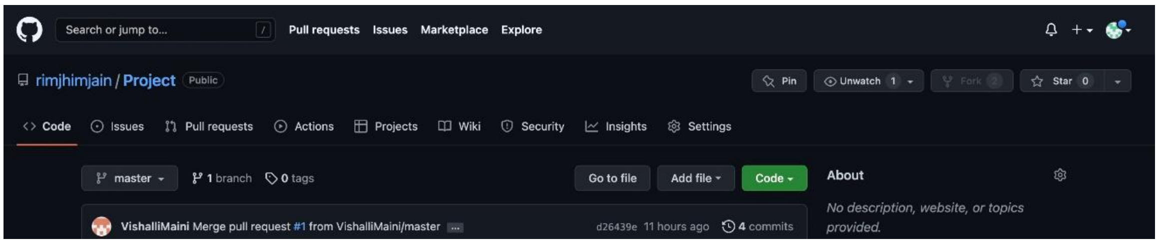
For these reasons, you are always suggested to FORK. Let's have a screenshot walkthrough of the whole process. When getting started with a contribution to Open Source Project, you have been advised to first FORK the repository(repo). But what is a fork?

You must have seen this icon on every repository in the top right corner. Now, this button is used to Fork the repo. But again, what is a fork or forking a repository in GitHub as shown in the below media as follows:



2.1.1 Procedure:

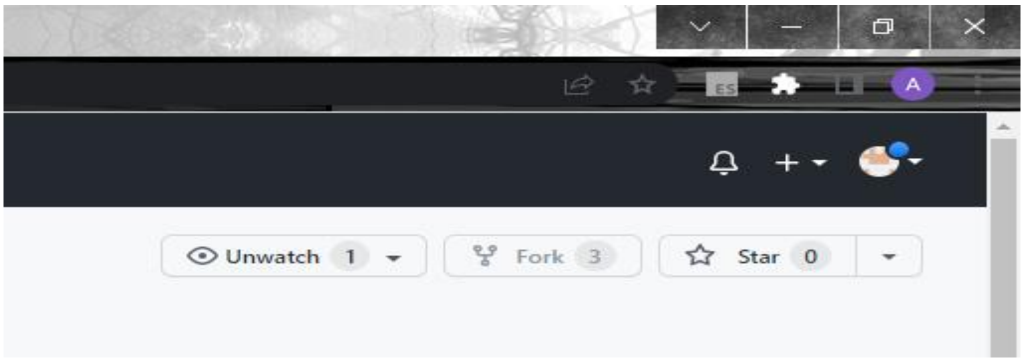
Step 1: Go to Projectwork official repository.



You can see rimjhimjain/projectwork This means rimjhimjain is the maintainer and Projectwork is the project's name.

Step 2: Find the Fork button on the top right corner.

You will see this screen.



Step 3: Click on Fork.

Step 4: Now you have your own copy of the repository.

But how can we confirm for which do refer to below visual aid as follows:

Now we can see the prachisingla12/projectwork also below that, we have the link to the original project I forked from

Whatever changes are made to ' prachisingla12/Projectwork ' will now change the original rimjhimjain/ProjectWork. We can make my changes here and then make a Pull Request to the maintainers of the project. Now it is in their hand if they will accept or reject your changes to the main project.

After this, we will see how to work in the forked repositories which were forked by the collaborators. If the collaborator tries to change in his forked repository, it will not affect the main repository.

2.2 What is COMMIT in GitHub?

Commit is like a snapshot of your repository. These commits are snapshots of your entire repository at specific times. You should make new commits often, based around logical units of change.
Over time, commits should tell a story of the history of your repository and how it came to be the way that it currently is.
Commits include lots of metadata in addition to the contents and message, like the author, timestamp and more.

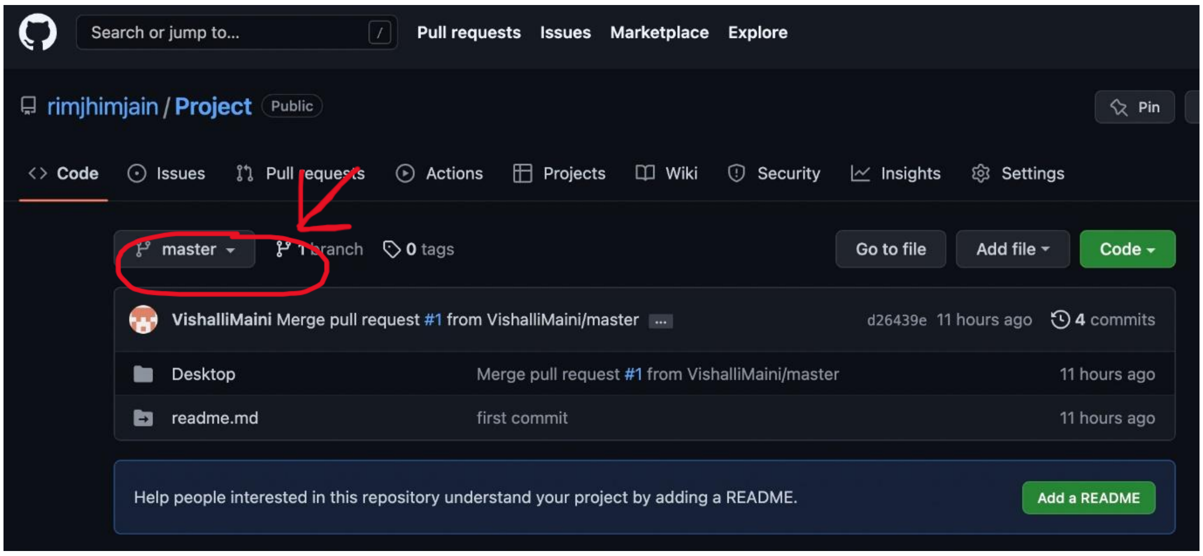
It is similar to saving a file that's been edited, a commit records changes to one or more files in your branch. Git assigns each commit a unique ID, called a SHA or hash, that identifies:

- The Specific Changes

- When the Changes were made
- Who created the changes

When you make a commit, you must include a commit message that briefly describes the changes, you can also add a co-author on any commits you collaborate on.

Now, we will see the main repository content of the main Projectwork Repository. We can see that there is a README.md file and other code files

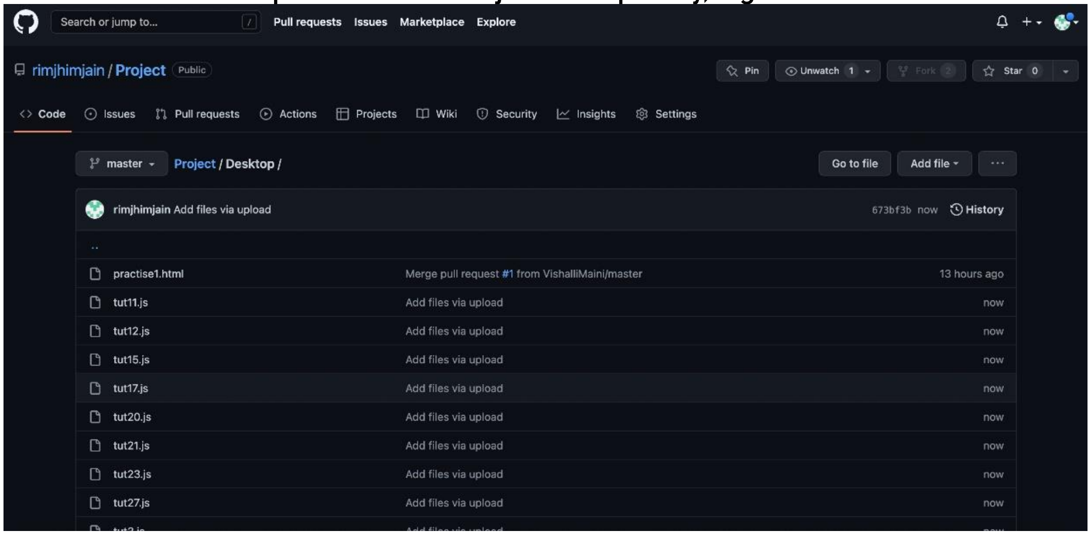


VishalliMaini Merge pull request #1 from VishalliMaini/master ... d26439e 11 hours ago 4 commits

Desktop	Merge pull request #1 from VishalliMaini/master	11 hours ago
readme.md	first commit	11 hours ago

Help people interested in this repository understand your project by adding a README.

The content in the Desktop file in the main Projectwork repository, is given as below.

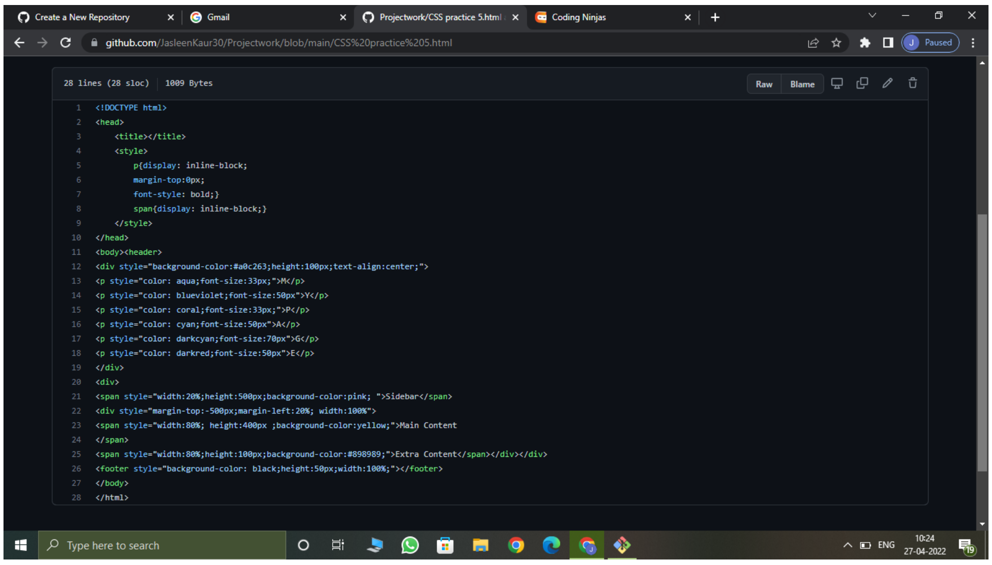


rimjhimjain Add files via upload 673bf3b now

..		
practise1.html	Merge pull request #1 from VishalliMaini/master	13 hours ago
tut11.js	Add files via upload	now
tut12.js	Add files via upload	now
tut15.js	Add files via upload	now
tut17.js	Add files via upload	now
tut20.js	Add files via upload	now
tut21.js	Add files via upload	now
tut23.js	Add files via upload	now
tut27.js	Add files via upload	now

Now, we will see the prachisingla12/Projectwork repository. We know that this repository was forked from rimjhimjain/Projectwork , so the content of this repository should have same files, and we can see that it also has the README.md file with other code files and after opening it has the same content as that of the main repository file.

- Now, when collaborator tries to change the content of any of the file in his forked repository i.e.,

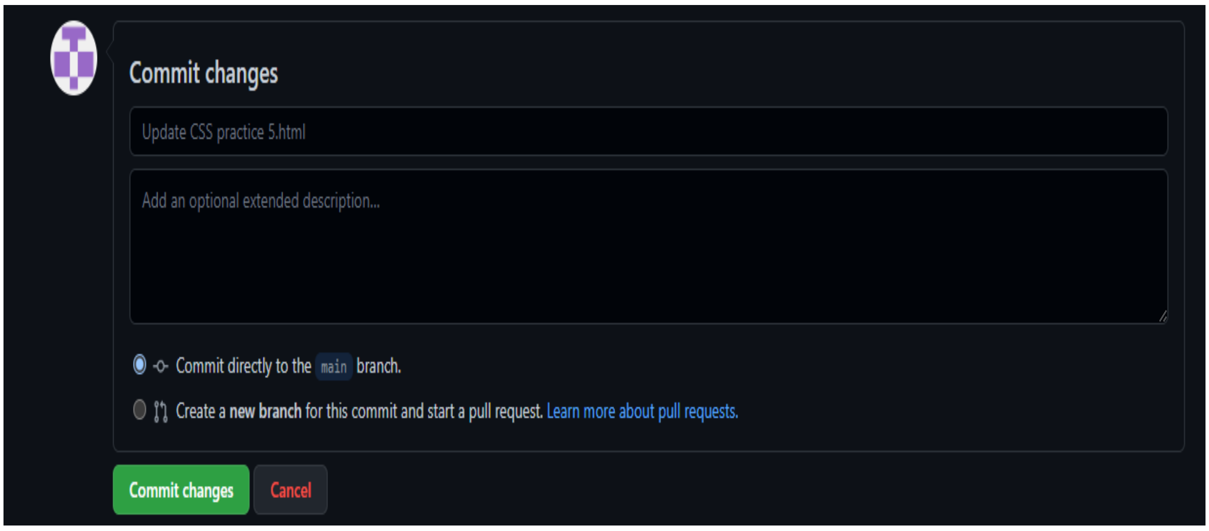


github.com/JasleenKaur30/Projectwork/blob/main/CSS%20practice%205.html

28 lines (28 sloc) | 1009 Bytes

```html
<!DOCTYPE html>
<head>
    <title></title>
    <style>
        p{display: inline-block;
        margin-top:0px;
        font-style: bold;}
        span{display: inline-block;}
    </style>
</head>
<body><header>
<div style="background-color:#a0c263;height:100px;text-align:center;">
<p style="color: aqua;font-size:33px;">M</p>
<p style="color: blueviolet;font-size:50px">Y</p>
<p style="color: coral;font-size:33px;">P</p>
<p style="color: cyan;font-size:50px">A</p>
<p style="color: darkcyan;font-size:70px">G</p>
<p style="color: darkred;font-size:50px">E</p>
</div>
<div>
<span style="width:20%;height:500px;background-color:pink; ">Sidebar</span>
<div style="margin-top:-500px;margin-left:20%; width:100%">
<span style="width:80%; height:400px ;background-color:yellow;">Main Content
</span>
<span style="width:80%;height:100px;background-color:#898989;">Extra Content</span></div></div>
<footer style="background-color: black;height:50px;width:100%;"></footer>
</body>
</html>
```

- We can see that the file has been edited and now it will be committed in this forked repository.



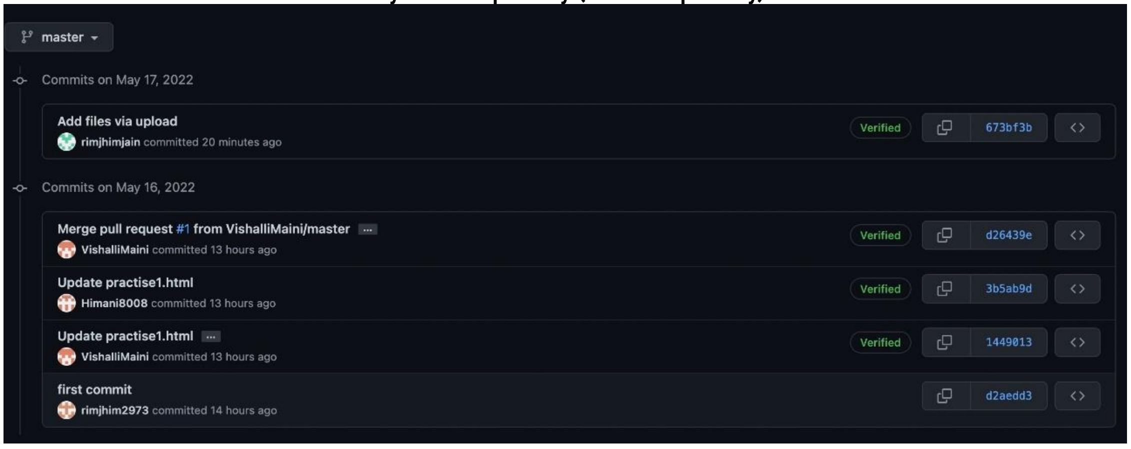
Commit changes

Update CSS practice 5.html

Add an optional extended description...

- [✓] Commit directly to the main branch.
- [ ] Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

- We can see the Commit history of the Repository (Forked Repository).



master

Commits on May 17, 2022

Add files via upload — rimjhimjain committed 20 minutes ago	Verified	673bf3b

Commits on May 16, 2022

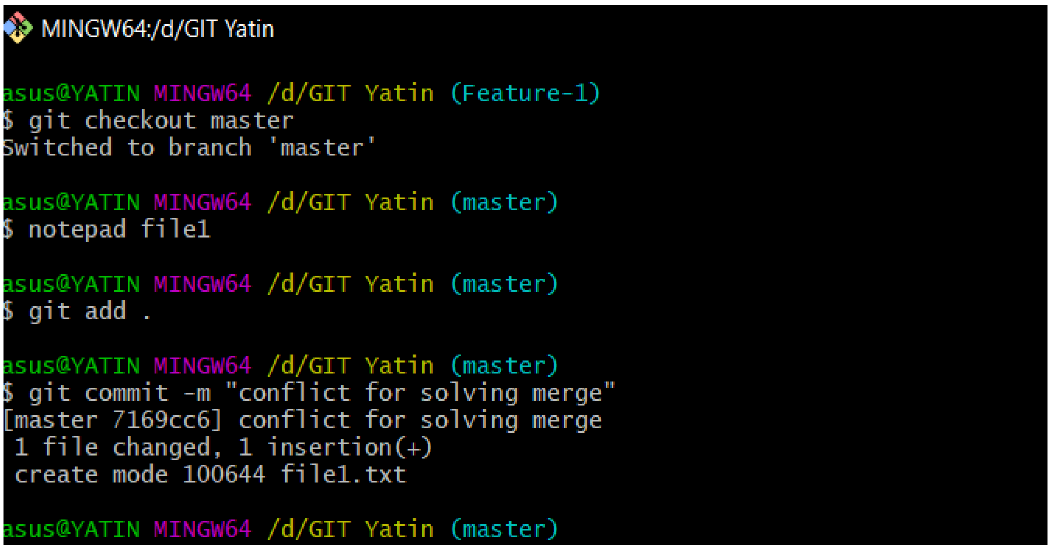
Merge pull request #1 from VishalliMaini/master … — VishalliMaini committed 13 hours ago	Verified	d26439e
Update practise1.html — Himani8008 committed 13 hours ago	Verified	3b5ab9d
Update practise1.html … — VishalliMaini committed 13 hours ago	Verified	1449013
first commit — rimjhim2973 committed 14 hours ago		d2aedd3

EXPERIMENT- 3

AIM: Merge and Resolve conflicts created due to ownactivity and collaborators activity.

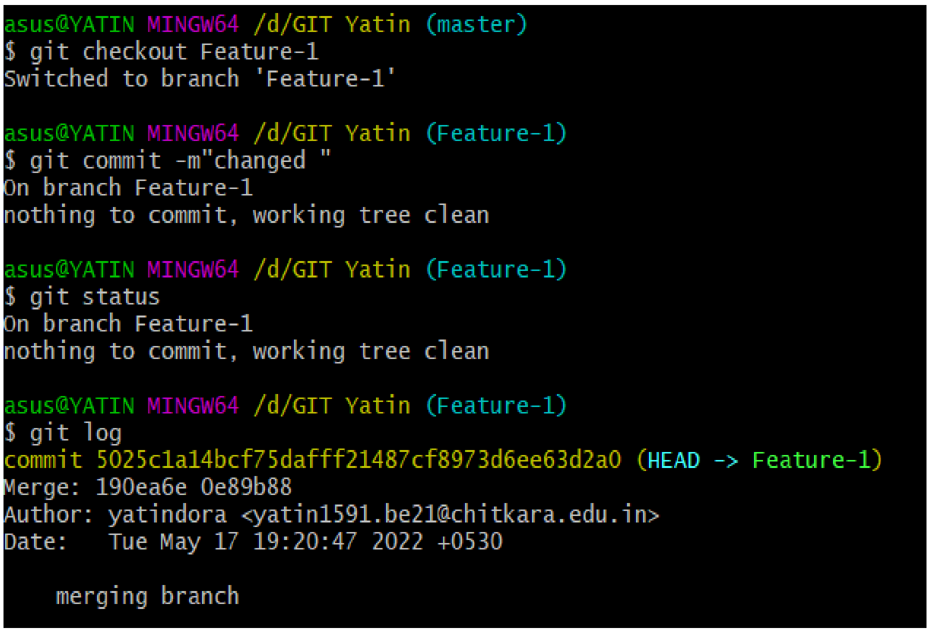
1. Do changes in master branch and commit those change. And checkout to "Feature-1" branch and again do changes and commit it. Now checkout to master branch and merge the Feature-1 branch in master.

Commit in Master Branch



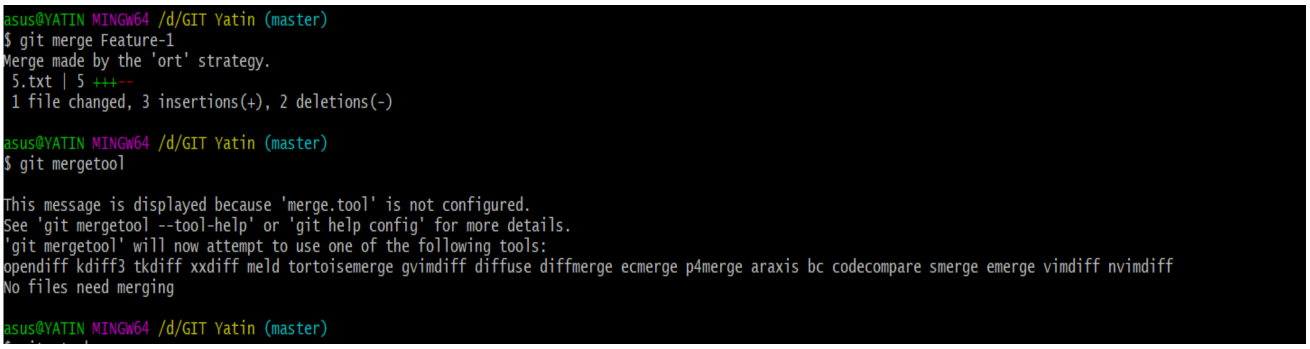
```
MINGW64:/d/GIT Yatin

asus@YATIN MINGW64 /d/GIT Yatin (Feature-1)
$ git checkout master
Switched to branch 'master'

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ notepad file1

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git add .

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git commit -m "conflict for solving merge"
[master 7169cc6] conflict for solving merge
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt

asus@YATIN MINGW64 /d/GIT Yatin (master)
```

Commit in Feature-1 Branch



```
asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git checkout Feature-1
Switched to branch 'Feature-1'

asus@YATIN MINGW64 /d/GIT Yatin (Feature-1)
$ git commit -m"changed "
On branch Feature-1
nothing to commit, working tree clean

asus@YATIN MINGW64 /d/GIT Yatin (Feature-1)
$ git status
On branch Feature-1
nothing to commit, working tree clean

asus@YATIN MINGW64 /d/GIT Yatin (Feature-1)
$ git log
commit 5025c1a14bcf75dafff21487cf8973d6ee63d2a0 (HEAD -> Feature-1)
Merge: 190ea6e 0e89b88
Author: yatindora <yatin1591.be21@chitkara.edu.in>
Date:   Tue May 17 19:20:47 2022 +0530

    merging branch
```

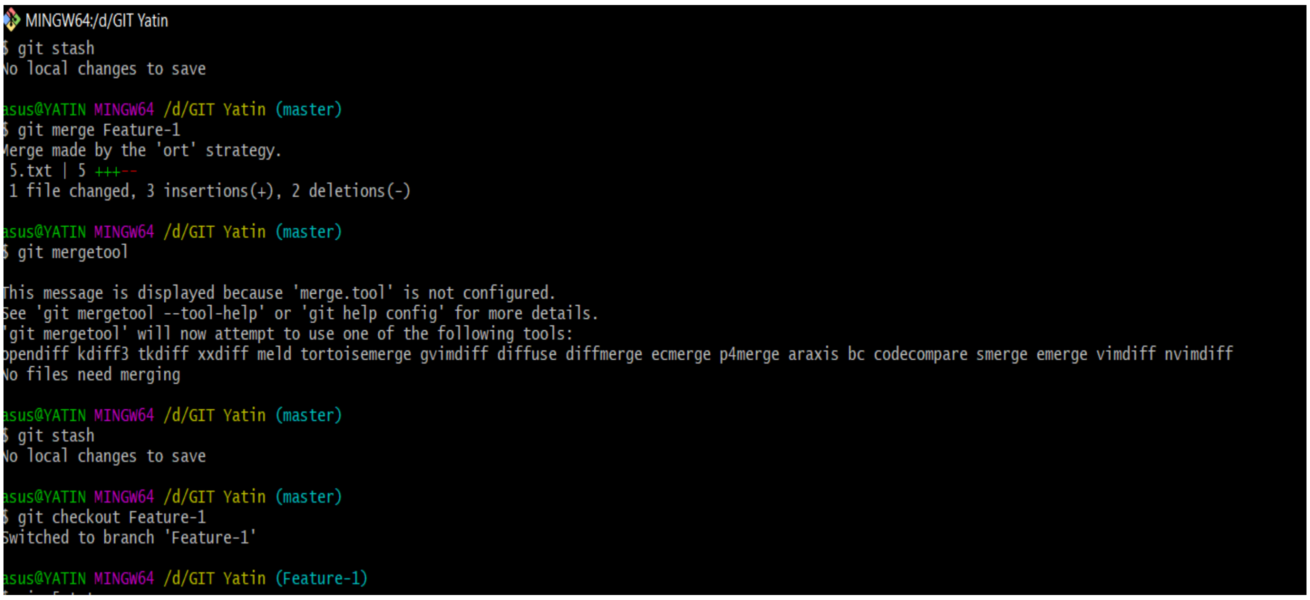
Due To My Activities , faces conflict during Merging



```
asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git merge Feature-1
Merge made by the 'ort' strategy.
 5.txt | 5 +++--
 1 file changed, 3 insertions(+), 2 deletions(-)

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff
No files need merging

asus@YATIN MINGW64 /d/GIT Yatin (master)
```

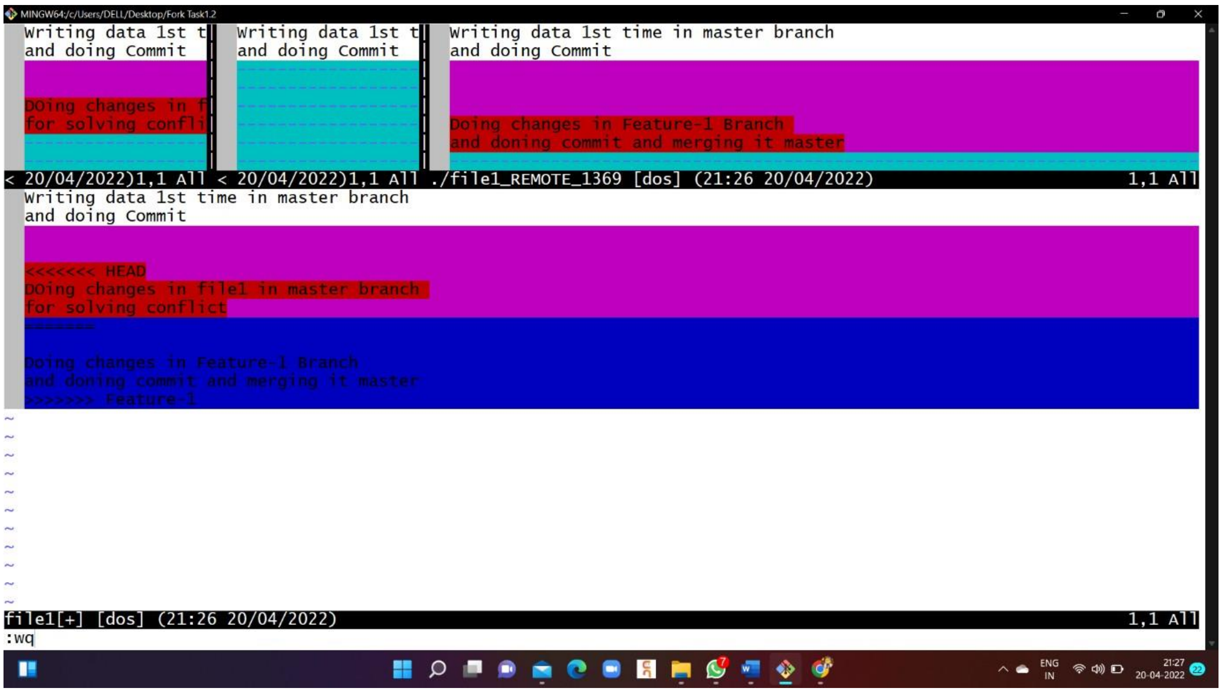
2. Use Command “Git mergetool” to solve the conflict

git-mergetool - Run merge conflict resolution tools to resolve merge conflicts

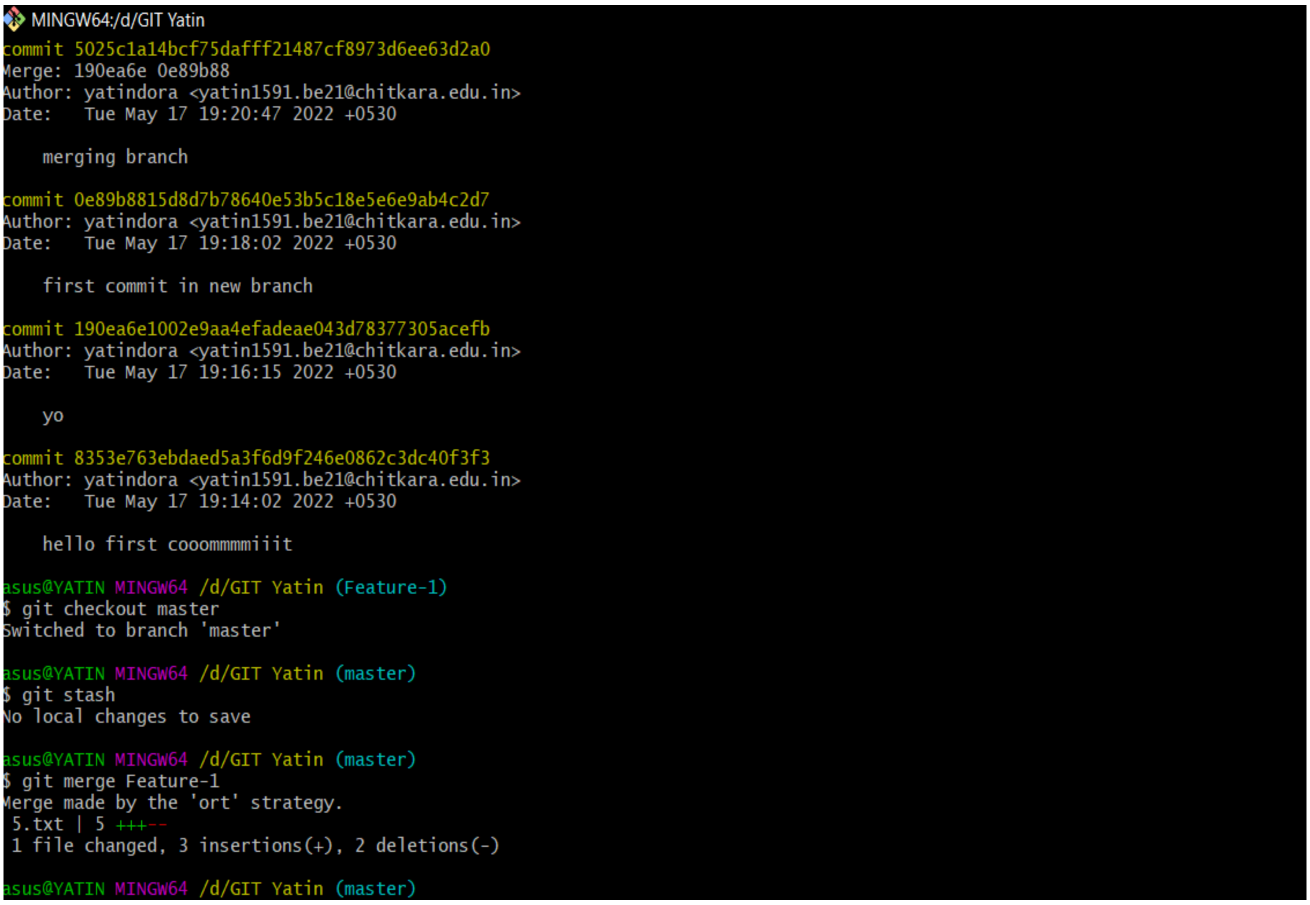


```
MINGW64:/d/GIT Yatin
$ git stash
No local changes to save

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git merge Feature-1
Merge made by the 'ort' strategy.
 5.txt | 5 +++--
 1 file changed, 3 insertions(+), 2 deletions(-)

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff
No files need merging

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git stash
No local changes to save

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git checkout Feature-1
Switched to branch 'Feature-1'

asus@YATIN MINGW64 /d/GIT Yatin (Feature-1)
```

2. Press “I” to insert, after insertion. Press “**:wq**”. The merge conflict is solved and our Feature-1 branch is merged to master branch.



```
MINGW64:/c/Users/DELL/Desktop/Fork Task1.2
Writing data 1st t  Writing data 1st t  Writing data 1st time in master branch
and doing Commit    and doing Commit    and doing Commit

Doing changes in f                      Doing changes in Feature-1 Branch
for solving confli                      and doning commit and merging it master
< 20/04/2022)1,1 All < 20/04/2022)1,1 All ./file1_REMOTE_1369 [dos] (21:26 20/04/2022)   1,1 All
Writing data 1st time in master branch
and doing Commit

<<<<<<< HEAD
Doing changes in file1 in master branch
for solving conflict
=======
Doing changes in Feature-1 Branch
and doning commit and merging it master
>>>>>>> Feature-1
~
~
~
~
~
~
~
~
~
~
~
file1[+] [dos] (21:26 20/04/2022)                                                        1,1 All
:wq
```



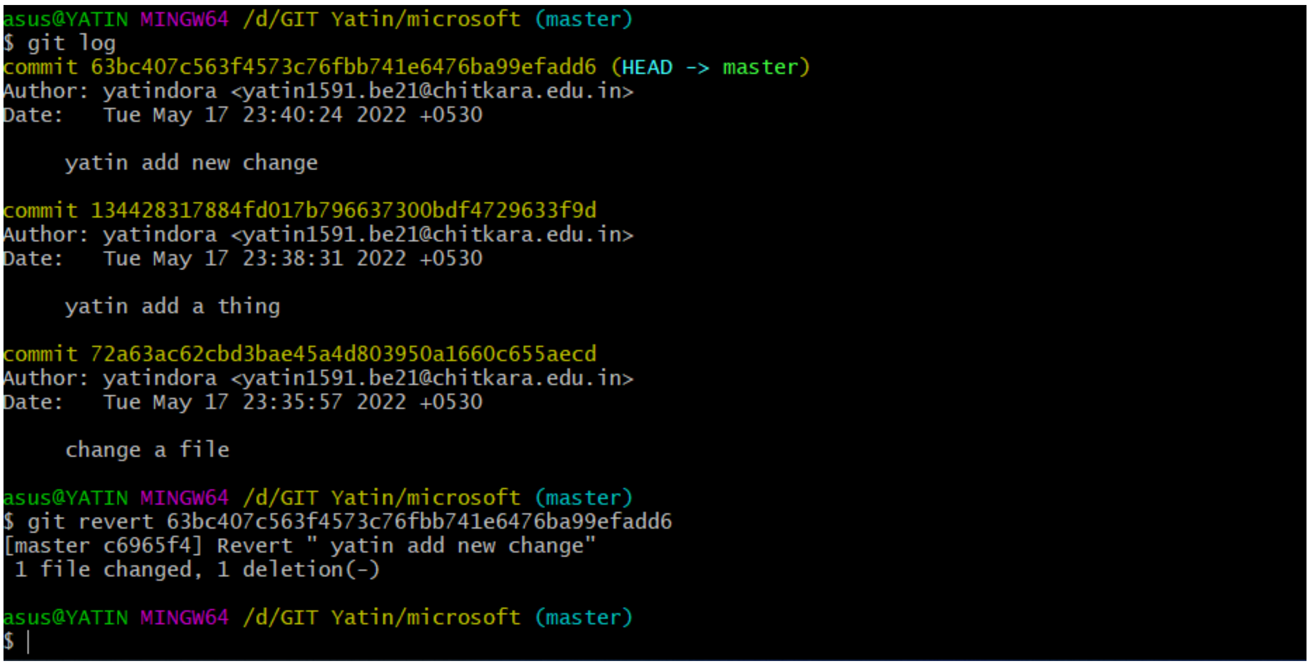
```
MINGW64:/d/GIT Yatin
commit 5025c1a14bcf75dafff21487cf8973d6ee63d2a0
Merge: 190ea6e 0e89b88
Author: yatindora <yatin1591.be21@chitkara.edu.in>
Date:   Tue May 17 19:20:47 2022 +0530

    merging branch

commit 0e89b8815d8d7b78640e53b5c18e5e6e9ab4c2d7
Author: yatindora <yatin1591.be21@chitkara.edu.in>
Date:   Tue May 17 19:18:02 2022 +0530

    first commit in new branch

commit 190ea6e1002e9aa4efadeae043d78377305acefb
Author: yatindora <yatin1591.be21@chitkara.edu.in>
Date:   Tue May 17 19:16:15 2022 +0530

    yo

commit 8353e763ebdaed5a3f6d9f246e0862c3dc40f3f3
Author: yatindora <yatin1591.be21@chitkara.edu.in>
Date:   Tue May 17 19:14:02 2022 +0530

    hello first cooommmmiiit

asus@YATIN MINGW64 /d/GIT Yatin (Feature-1)
$ git checkout master
Switched to branch 'master'

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git stash
No local changes to save

asus@YATIN MINGW64 /d/GIT Yatin (master)
$ git merge Feature-1
Merge made by the 'ort' strategy.
 5.txt | 5 +++--
 1 file changed, 3 insertions(+), 2 deletions(-)

asus@YATIN MINGW64 /d/GIT Yatin (master)
```

EXPERIMENT- 4

AIM : Reset and Revert

git-revert - Revert some existing commits.

1.On GitBash CLI , Type command “git Commad <Commit id>” . It revert the changes that done before Commit.



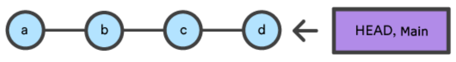
```
asus@YATIN MINGW64 /d/GIT Yatin/microsoft (master)
$ git log
commit 63bc407c563f4573c76fbb741e6476ba99efadd6 (HEAD -> master)
Author: yatindora <yatin1591.be21@chitkara.edu.in>
Date:   Tue May 17 23:40:24 2022 +0530

    yatin add new change

commit 134428317884fd017b796637300bdf4729633f9d
Author: yatindora <yatin1591.be21@chitkara.edu.in>
Date:   Tue May 17 23:38:31 2022 +0530

    yatin add a thing

commit 72a63ac62cbd3bae45a4d803950a1660c655aecd
Author: yatindora <yatin1591.be21@chitkara.edu.in>
Date:   Tue May 17 23:35:57 2022 +0530

    change a file

asus@YATIN MINGW64 /d/GIT Yatin/microsoft (master)
$ git revert 63bc407c563f4573c76fbb741e6476ba99efadd6
[master c6965f4] Revert " yatin add new change"
 1 file changed, 1 deletion(-)

asus@YATIN MINGW64 /d/GIT Yatin/microsoft (master)
$ |
```

`git revert HEAD~3 :-`

Revert the changes specified by the fourth last commit in HEAD and create a new commit with the reverted changes.

git-reset - Reset current HEAD to the specified state.

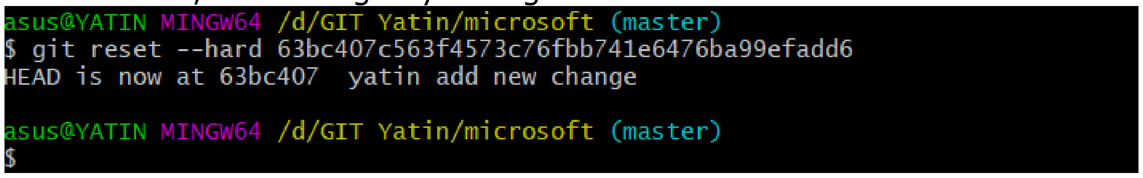
At a surface level, `git reset` is similar in behavior to `git checkout`. Where `git checkout` solely operates on the `HEAD` ref pointer, `git` `reset` will move the `HEAD` ref pointer and the current branch ref pointer. To better demonstrate this behavior consider the following example:



This example demonstrates a sequence of commits on the `main` branch. The `HEAD` ref and `main` branch ref currently point to commit d. Now let us execute and compare, both `git checkout b` and `git reset b`.

Git Reset

`reset` is the command we use when we want to move the repository back to a previous `commit`, discarding any changes made after that `commit`.



```
asus@YATIN MINGW64 /d/GIT Yatin/microsoft (master)
$ git reset --hard 63bc407c563f4573c76fbb741e6476ba99efadd6
HEAD is now at 63bc407  yatin add new change

asus@YATIN MINGW64 /d/GIT Yatin/microsoft (master)
$
```

SUMMARY

The tasks performed in all are listed below:

1. Add collaborators on GitHub Repo
2. Fork and Commit
3. Merge and Resolve conflicts created due to own activity and collaborators activity.
4. Reset and Revert