

Subject Name: Source Code Management

Subject Code: CS181

Cluster: Gamma

Department: CSE

CHITKARA
UNIVERSITY



Submitted By:

Mehvish Aggarwal

2110990882

G-12

Submitted To:

Ridhima Rani

Source Code Management (Task 1.2)

S. No	Task Title	Page No.
1.	<u>Add collaborators on GitHub Repo</u>	3 - 6
2.	<u>Fork and Commit</u>	7 - 8
3.	<u>Merge and Resolve conflicts created due to own activity and collaborators activity.</u>	9 - 11
4.	<u>Reset and Revert</u>	12- 13

EXPERIMENT: 1

AIM: Add collaborators on GitHub Repo

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: mehvish002 / Repository name: experiment-1.2

Great repository names are short and memorable. Need inspiration? How about [fantastic-waddle](#)?

Description (optional)

☐ 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.](#)
.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)
License: None

This will set `main` as the default branch. Change the default name in your settings.

☐ You are creating a private repository in your personal account.

[Create repository](#)

2. Now Copy the HTTP link of repo and paste it on your 'Git BASH', and merge the local repo in remote repo (i.e.) experiment 1.2.

mehvish002 / experiment-1.2

Initial commit

README.md

experiment-1.2

```

MINGW64 ~/OneDrive/Desktop/git (master)
$ git log
commit 74089622180ef2c8504a23326208ee9de53b759 (HEAD -> master, activity1)
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 05:05:33 2022 +0530

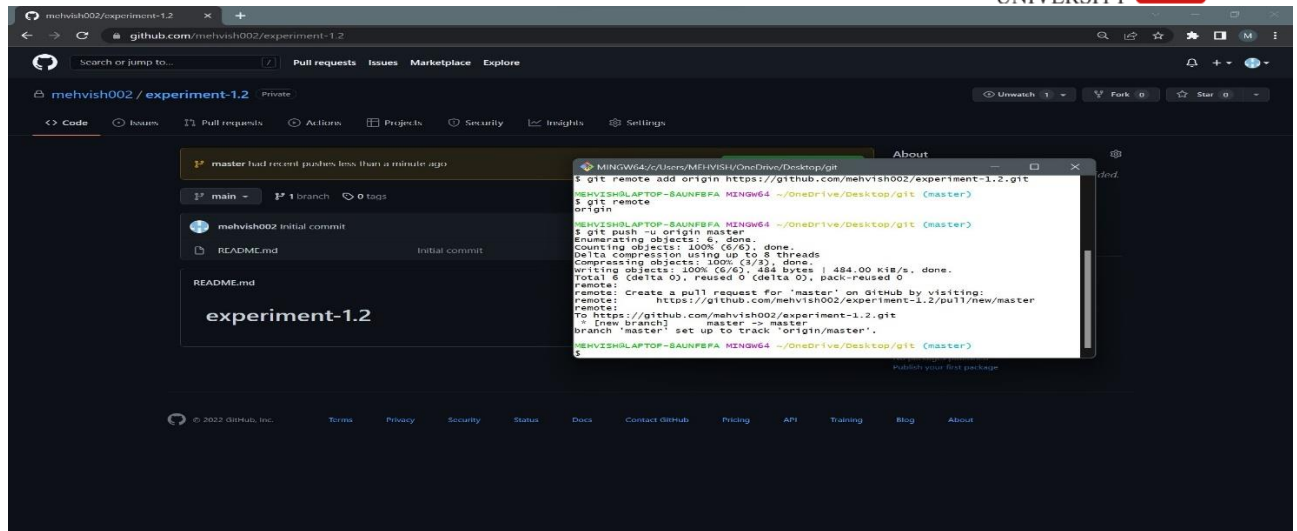
    this is file1

commit b3676cfbe54ef00522cbf623780f01865e8d6ab4
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 05:03:32 2022 +0530

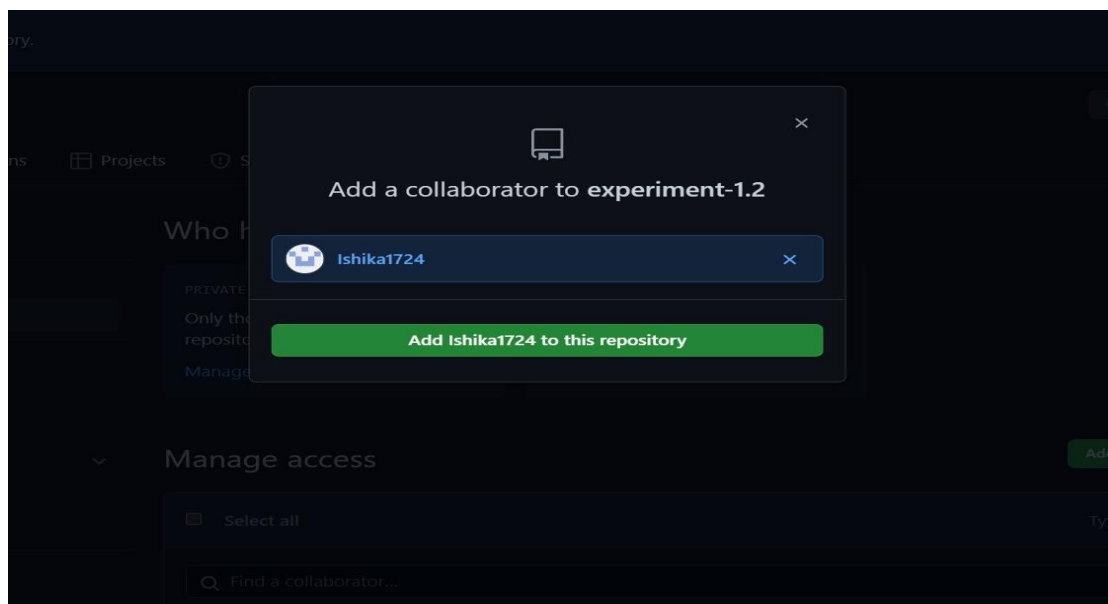
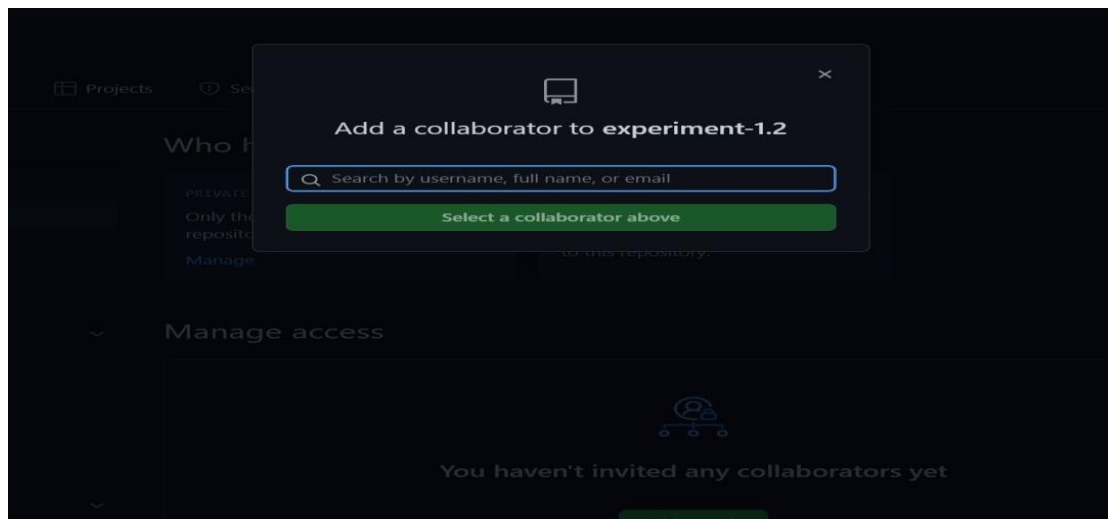
    my first file

MINGW64 ~/OneDrive/Desktop/git (master)
$ git remote add origin https://github.com/mehvish002/experiment-1.2.git
$ git remote -v
origin
https://github.com/mehvish002/experiment-1.2.git (fetch)
https://github.com/mehvish002/experiment-1.2.git (push)
MINGW64 ~/OneDrive/Desktop/git (master)
$ git push origin master

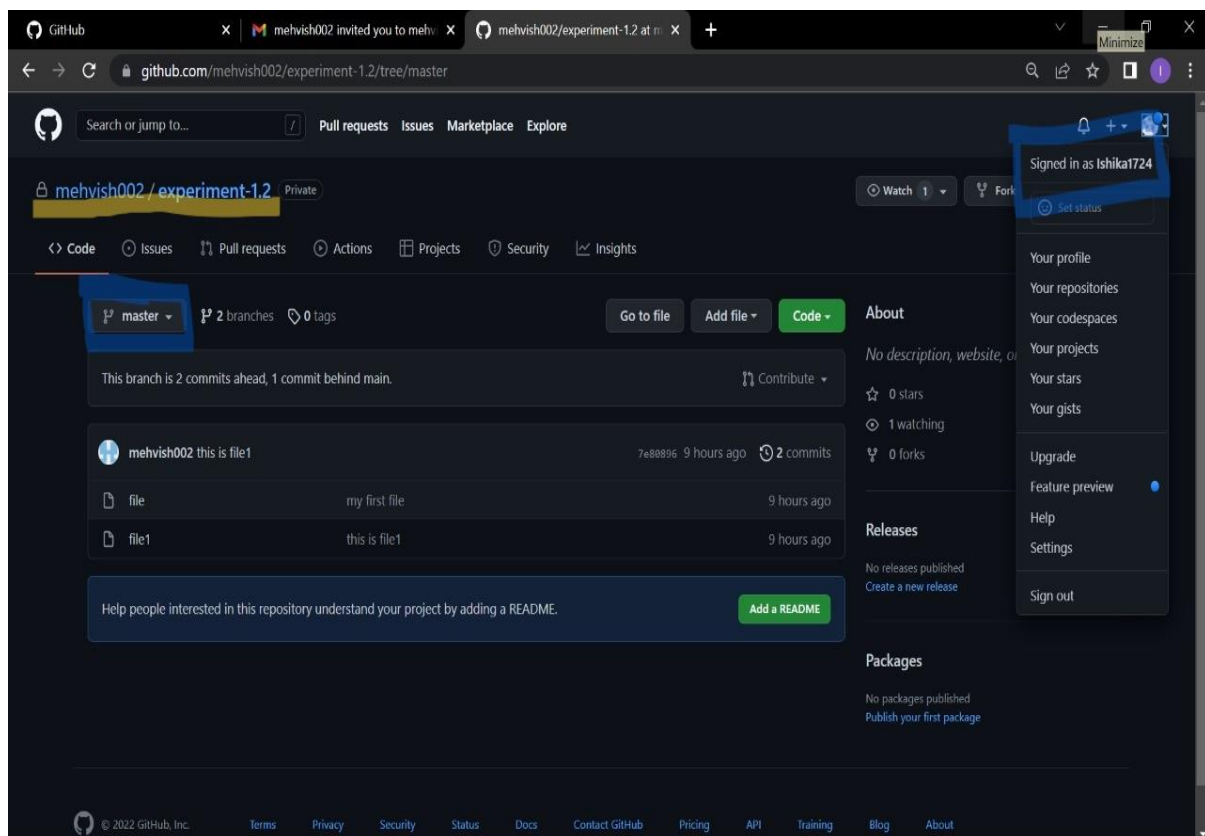
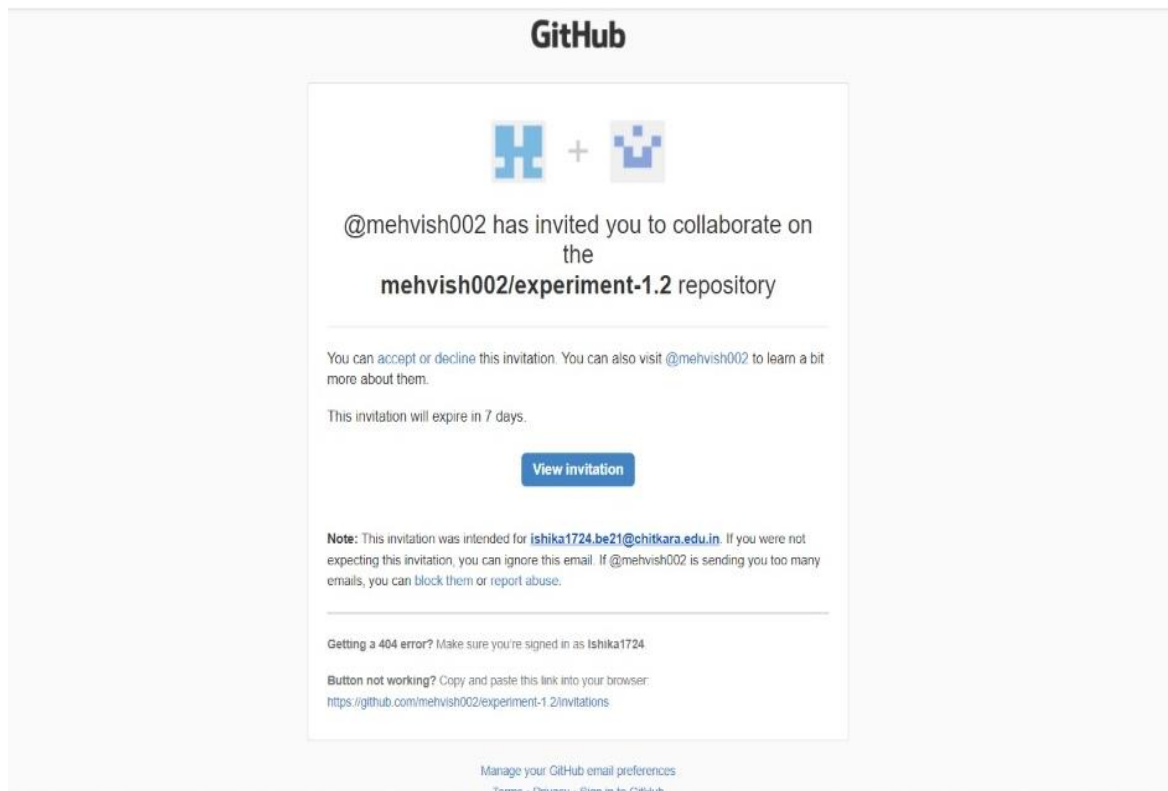
```



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



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



5. New Collaborator has now access to the experiment (1.2).

The screenshot shows the GitHub repository settings for 'mehvish002/experiment-1.2'. The 'Collaborators' tab is selected in the left sidebar. The 'Who has access' section shows 'DIRECT ACCESS' with 2 collaborators and 2 invitations. The 'Manage access' section lists two collaborators: Anushkajain22 and Ishika1724, both with 'Pending invite' status. A yellow box highlights the 'Manage access' section, and a blue box highlights the 'DIRECT ACCESS' summary.

Who has access

PRIVATE REPOSITORY
Only those with access to this repository can view it.
[Manage](#)

DIRECT ACCESS
2 have access to this repository, 0 collaborators, 2 invitations.

Manage access [Add people](#)

Select all	Type
Find a collaborator...	
<input type="checkbox"/> Anushkajain22 Awaiting Anushkajain22's response	Pending invite Remove
<input type="checkbox"/> Ishika1724 Awaiting Ishika1724's response	Pending invite Remove

< Previous Next >

EXPERIMENT: 2

AIM: Fork and Commit

Fork: - A Fork is copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original Project.

1. Create a New Repo (or Folder) and Initialize the Git repository by Using 'git init'. And add git origin of your remote Repo to local Repo.

```

MINGW64~/c:/Users/MEHVISH/OneDrive/Desktop/fork 1.2
$ git init
Initialized empty Git repository in C:/Users/MEHVISH/OneDrive/Desktop/fork 1.2/.
$ git remote add origin https://github.com/mehvish002/experiment-1.2.git
$ git remote -v
origin https://github.com/mehvish002/experiment-1.2.git (fetch)
origin https://github.com/mehvish002/experiment-1.2.git (push)

```

2. Type "Git pull https://github.com/mehvish002/experiment-1.2-.git" on CLI.

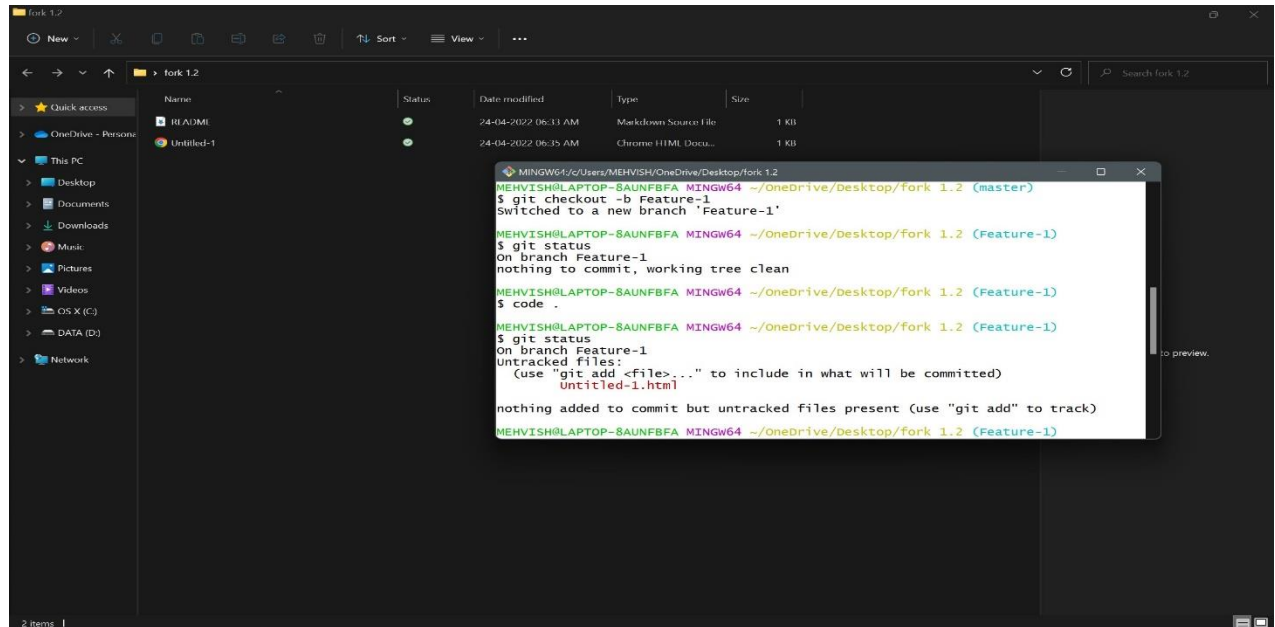
Git pull <url> → This command is used to fetch the remote repo or to clone the repo.

```

MINGW64~/c:/Users/MEHVISH/OneDrive/Desktop/fork 1.2
$ git pull https://github.com/mehvish002/experiment-1.2.git
remote: Enumerating objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 600 bytes | 15.00 KiB/s, done.
From https://github.com/mehvish002/experiment-1.2
* branch HEAD -> FETCH_HEAD
$ git log
commit cea4023534753761441dda7addc70ba820cb1f5b (HEAD -> master)
Author: mehvish002 <97453236+mehvish002@users.noreply.github.com>
Date: Sun Apr 24 04:47:49 2022 +0530

```

3. Create a new Branch "Feature-1". Open and file and do changes in it and commit it.



```

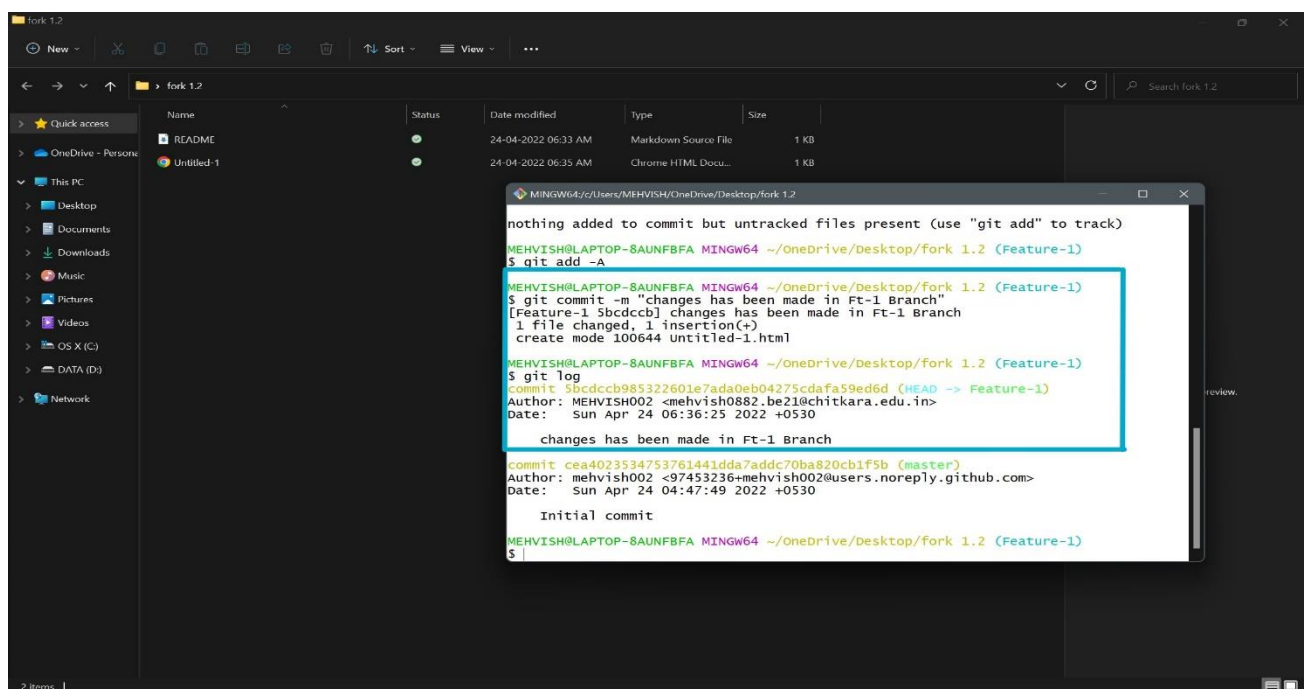
MINGW64~/c:/Users/MEHVISH/OneDrive/Desktop/fork 1.2
MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (master)
$ git checkout -b Feature-1
Switched to a new branch 'Feature-1'

MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$ git status
On branch Feature-1
nothing to commit, working tree clean

MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$ code .
MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$ git status
On branch Feature-1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Untitled-1.html

nothing added to commit but untracked files present (use "git add" to track)
MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)

```



```

nothing added to commit but untracked files present (use "git add" to track)
MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$ git add -A
MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$ git commit -m "changes has been made in Ft-1 Branch"
[Feature-1 5bcdccb] changes has been made in Ft-1 Branch
1 file changed, 1 insertion(+)
create mode 100644 Untitled-1.html

MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$ git log
commit 5bcdccb985322601e7ada0eb04275cdafa59ed6d (HEAD -> Feature-1)
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 06:36:25 2022 +0530

    changes has been made in Ft-1 Branch

commit cea4023534753761441dda7addc70ba820cb1f5b (master)
Author: mehvish002 <97453236+mehvish002@users.noreply.github.com>
Date: Sun Apr 24 04:47:49 2022 +0530

    Initial commit
MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$

```

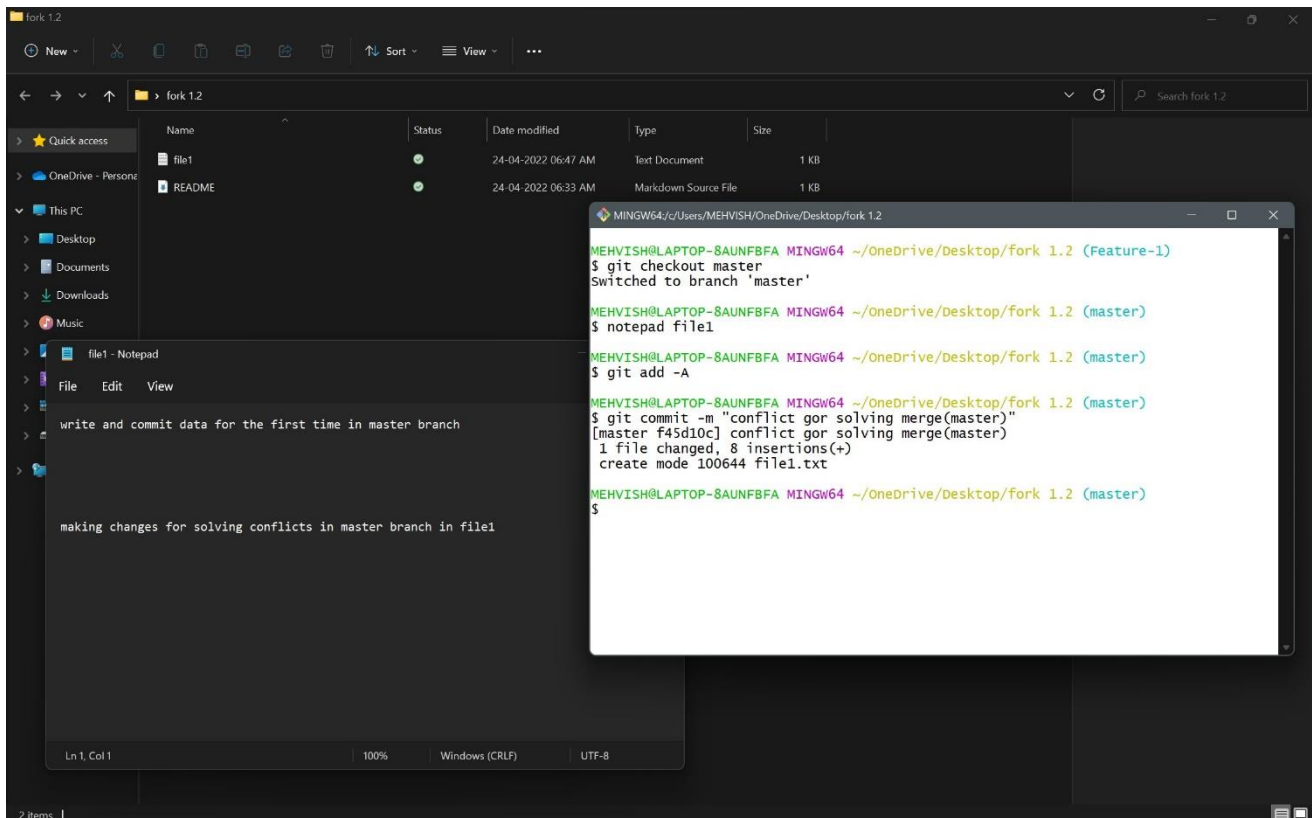
Commit is made on the fork Repo (i.e) Mehvish-Aggarwal-Experiment-1.2-

EXPERIMENT :3

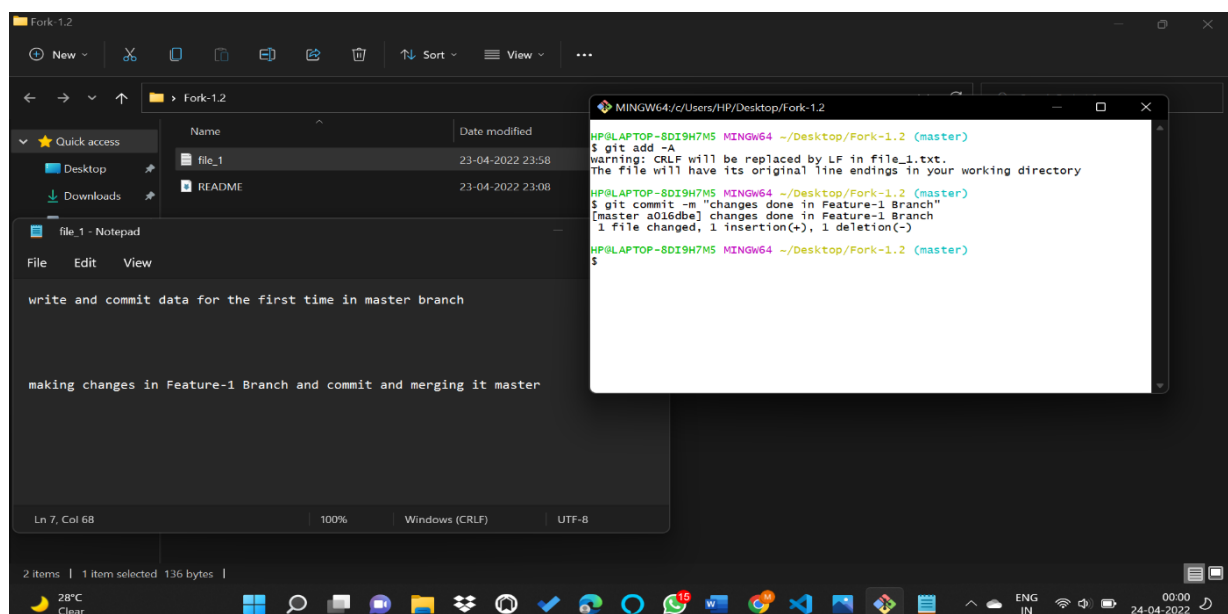
AIM: Merge and Resolve conflicts created due to own activity and collaborators activity.

1. Make changes in master branch and commit those changes, checkout to “Feature-1” branch and again make changes then commit them. Now checkout to master branch and merge the Feature-1 branch in master.

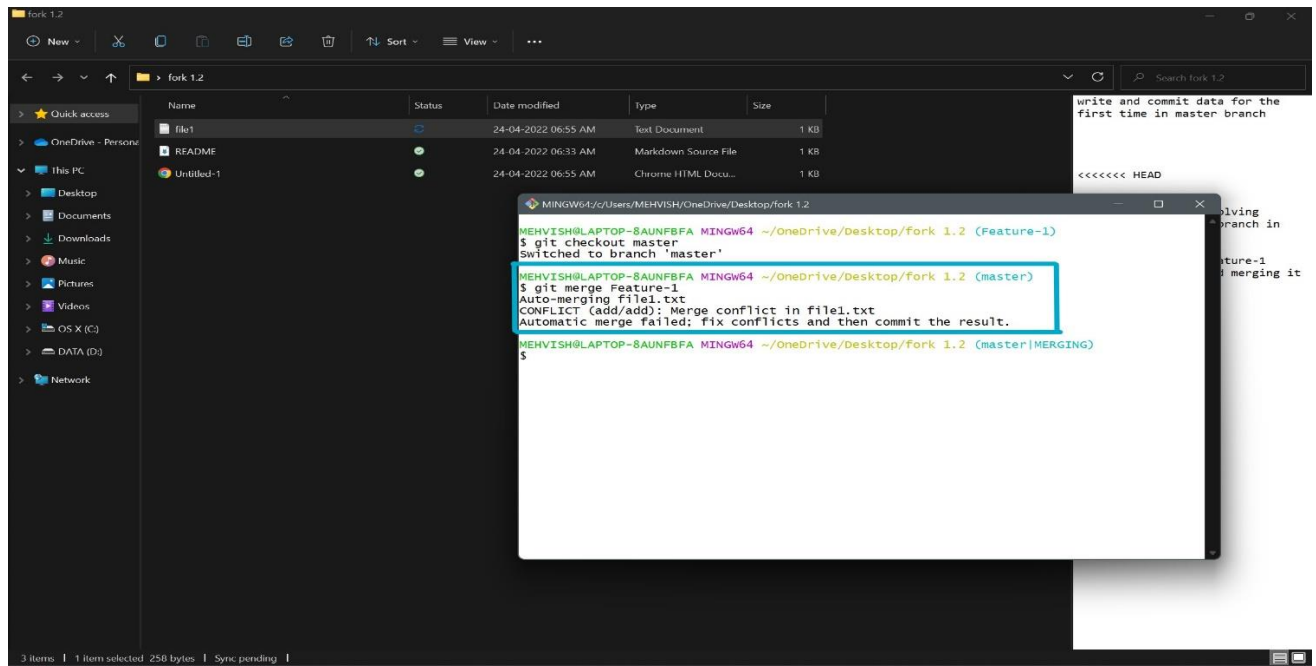
Commit in Master Branch



Commit in Feature-1 Branch

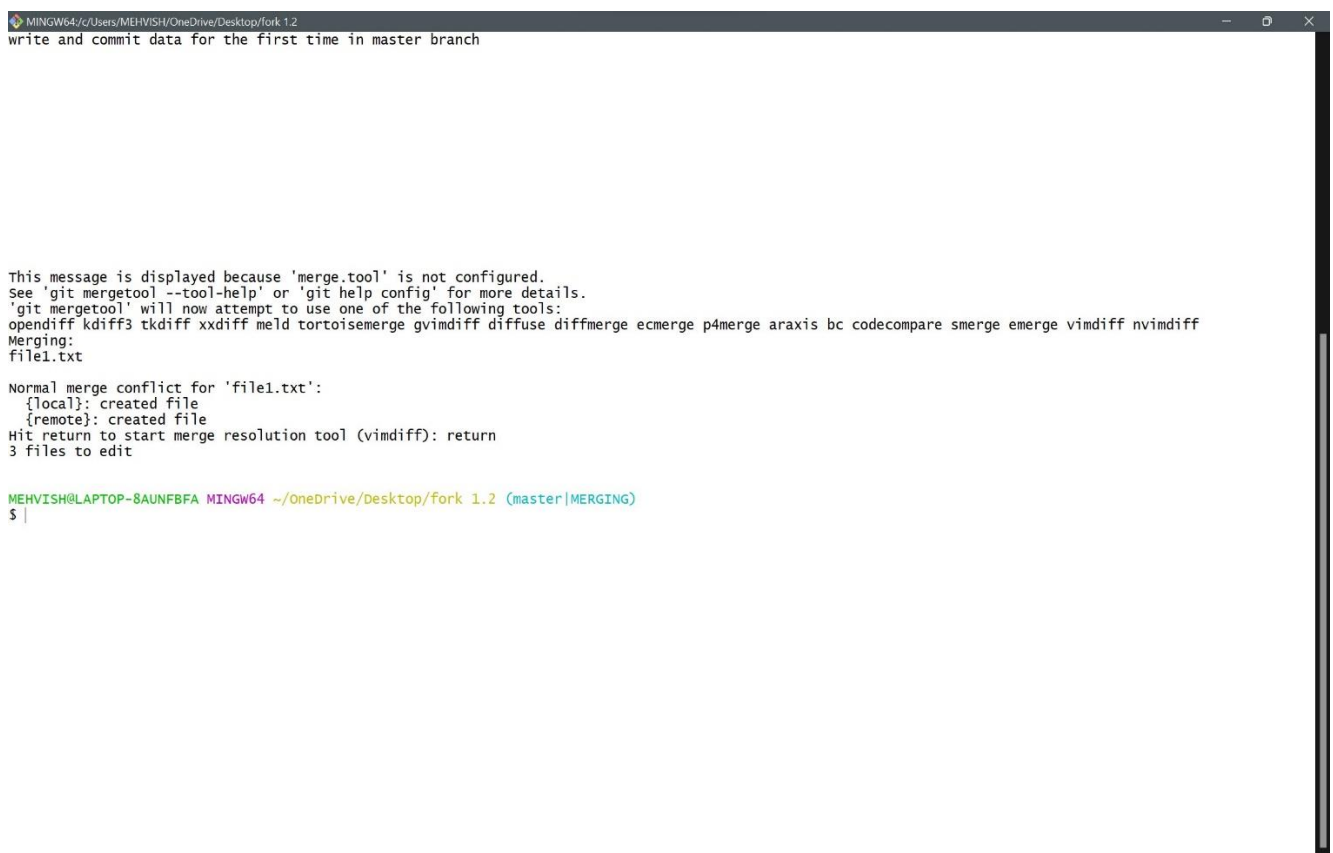


Due To My Activities , faces conflict during Merging

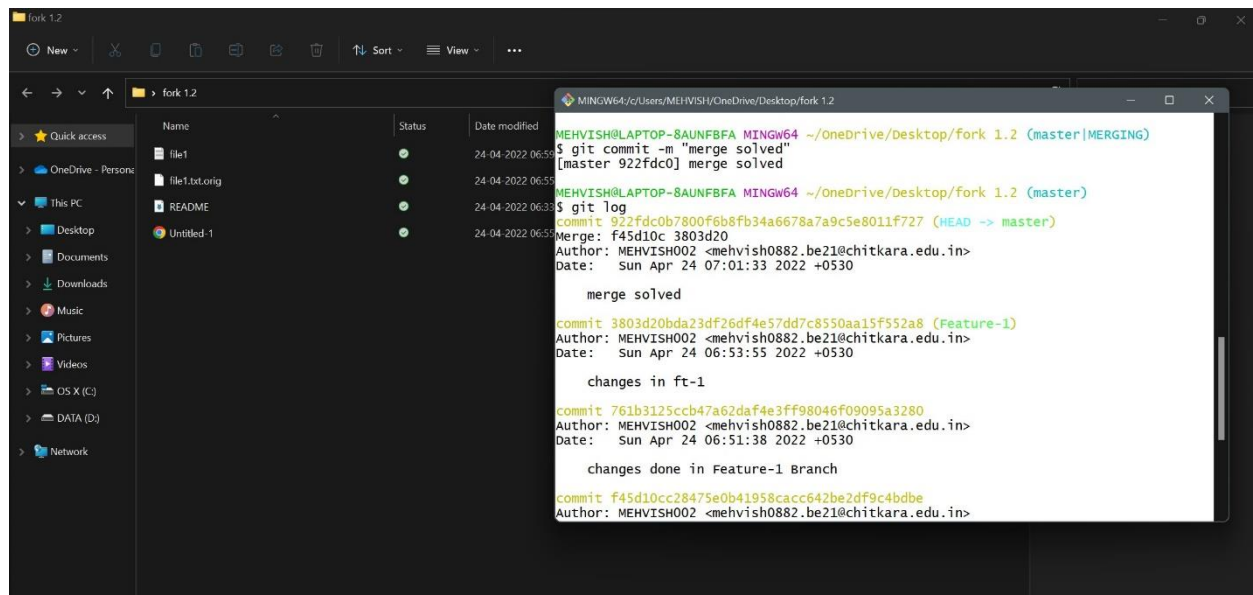
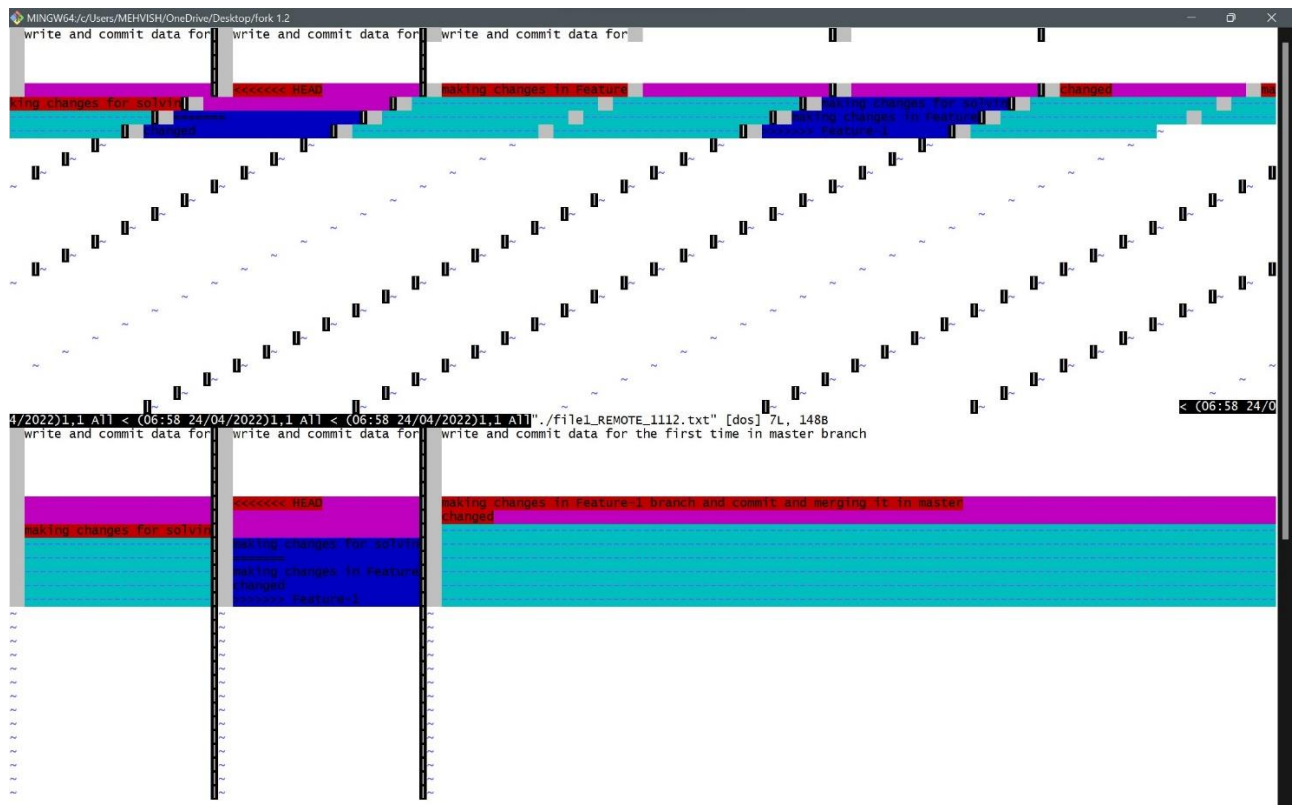


2. Use Command “Git mergetool” to solve the conflict

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



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



EXPERIMENT :4

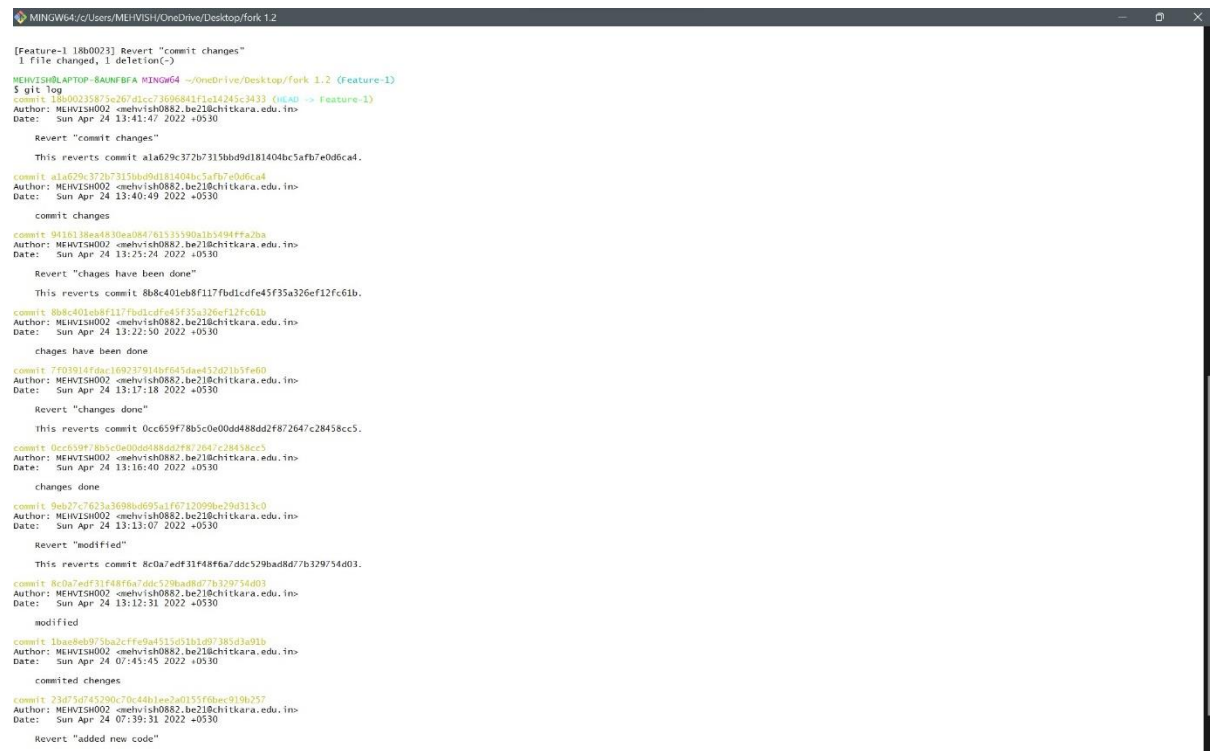
AIM: Reset and Revert

git-revert - Revert some existing commits.

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

```
MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$ git log
commit ala629c372b7315bbd9d181404bc5afb7e0d6ca4 (HEAD -> Feature-1)
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 13:40:49 2022 +0530

    commit changes
```



```

[Feature-1 18b0023] Revert "commit changes"
1 file changed, 1 deletion(-)

MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$ git log
commit ala629c372b7315bbd9d181404bc5afb7e0d6ca4 (HEAD -> Feature-1)
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 13:41:47 2022 +0530

    commit changes

Revert "commit changes"

This reverts commit ala629c372b7315bbd9d181404bc5afb7e0d6ca4.

commit ala629c372b7315bbd9d181404bc5afb7e0d6ca4
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 13:40:49 2022 +0530

    commit changes

Revert "changes have been done"

This reverts commit 8b8c401eb8f117fbd1cdf45f35a326ef12fc61b.

commit 8b8c401eb8f117fbd1cdf45f35a326ef12fc61b
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 13:22:50 2022 +0530

    changes have been done

Revert "changes done"

This reverts commit 0cc659f78b5c0e0d4d88dd2f872647c28458cc5.

commit 0cc659f78b5c0e0d4d88dd2f872647c28458cc5
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 13:16:40 2022 +0530

    changes done

Revert "modified"

This reverts commit 8c0a7edf31f48f6a7ddc529bad8d7b329754d03.

commit 8c0a7edf31f48f6a7ddc529bad8d7b329754d03
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 13:12:31 2022 +0530

    modified

commit 1b9eb4b073b4c3ff0b125d31a4d0738547a91b
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 07:45:45 2022 +0530

    committed changes

commit 23d75d745290c70c44b1ee2a0155f6bec919b257
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 07:39:31 2022 +0530

    Revert "added new code"
```

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 behaviour 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 behaviour 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`.

```
MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$ git reset --hard 1bae8eb975ba2cffe9a4515d51b1d97385d3a91b
HEAD is now at 1bae8eb committed changes
```

```
MEHVISH@LAPTOP-8AUNFBFA MINGW64 ~/OneDrive/Desktop/fork 1.2 (Feature-1)
$ git log
commit 1bae8eb975ba2cffe9a4515d51b1d97385d3a91b (HEAD -> Feature-1)
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 07:45:45 2022 +0530
```

committed changes

```
commit 23d75d745290c70c44b1ee2a0155f6bec919b257
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 07:39:31 2022 +0530
```

Revert "added new code"

This reverts commit 3a78a9839bdd3f9af14fd0aad83e48d259060e2f.

```
commit 3a78a9839bdd3f9af14fd0aad83e48d259060e2f
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 07:15:23 2022 +0530
```

added new code

```
commit 5c07e58a0d9c71200735ae0a80f290a4fa247abe
Author: MEHVISH002 <mehvish0882.be21@chitkara.edu.in>
Date: Sun Apr 24 07:14:22 2022 +0530
```

created new file