

Subject Name: Source Code Management

Subject Code: CS181

Cluster: Beta

Department: CSE

CHITKARA
UNIVERSITY



Submitted By:

Abhishek

2110990058

G01-B

Submitted To:

Dr. Monit Kapoor

List of Programs

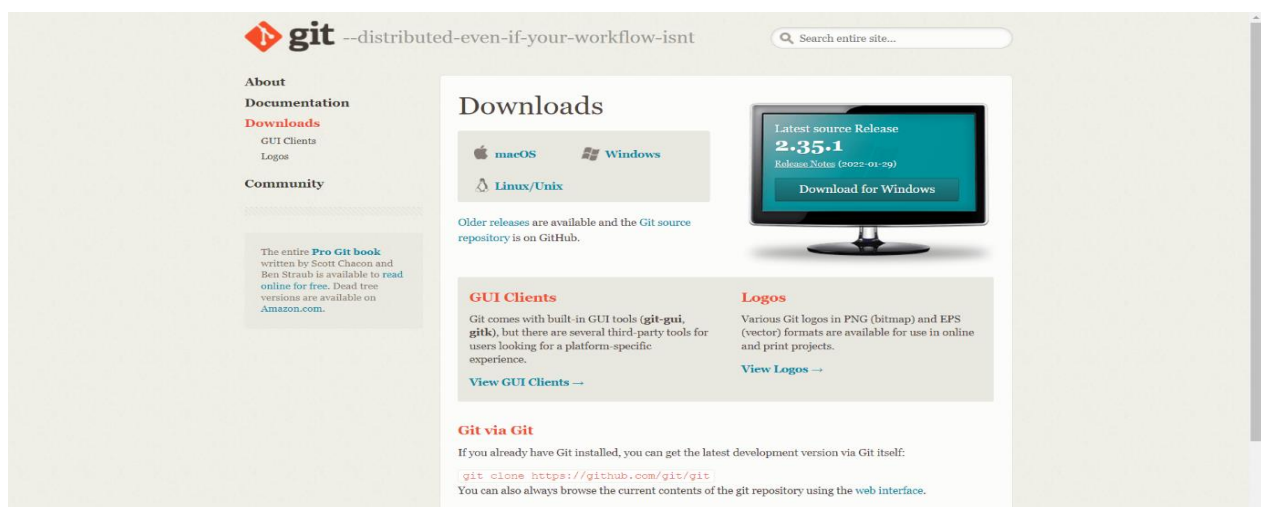
[illegible]

Experiment No. 01

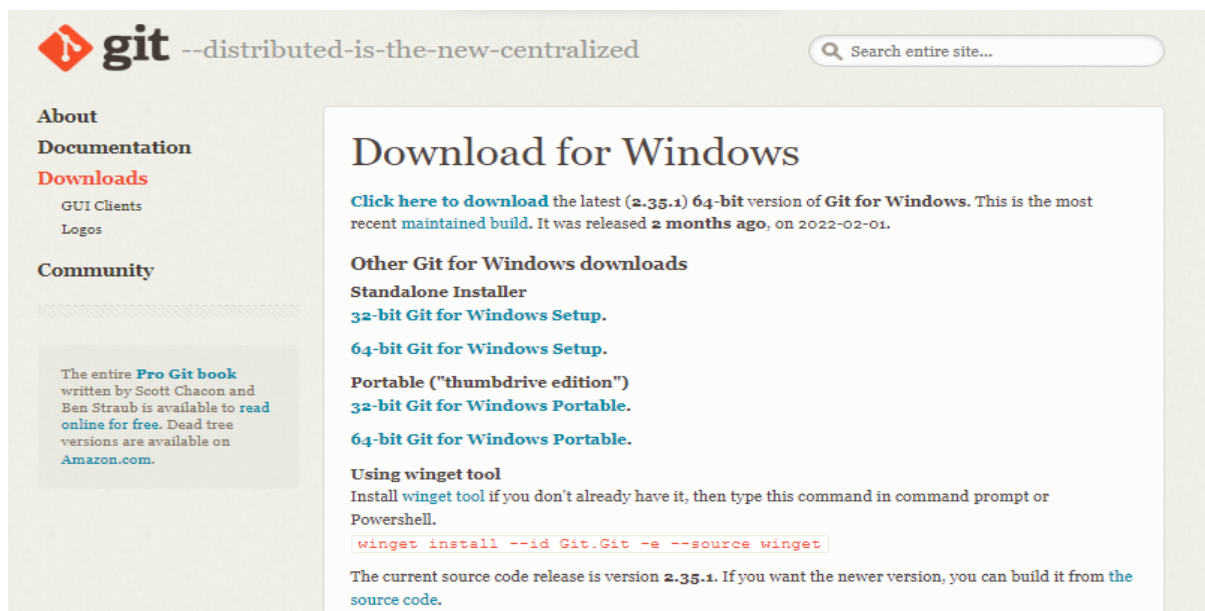
Aim: - Setting up of Git client

1. Visit the provided link for the installation of git in your system.

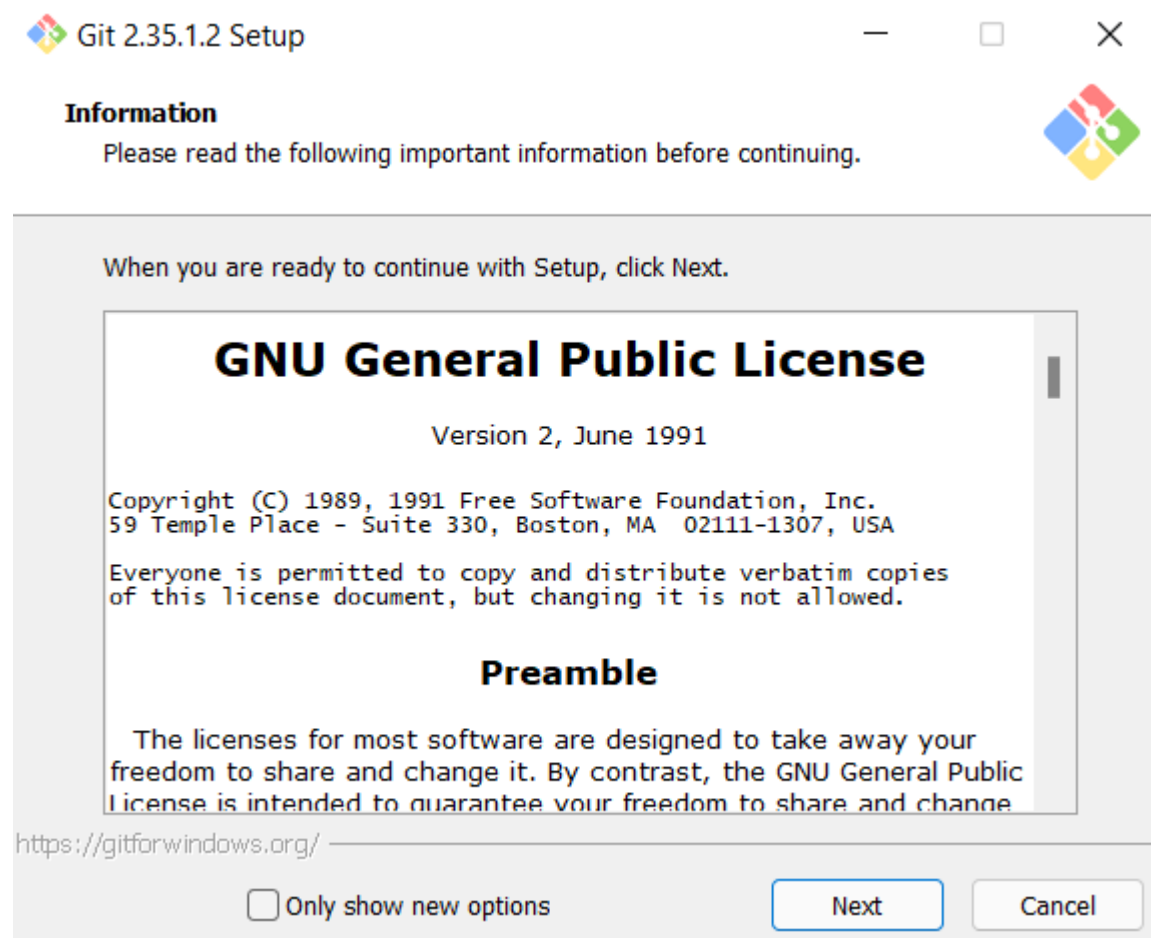
<https://git-scm.com/downloads>



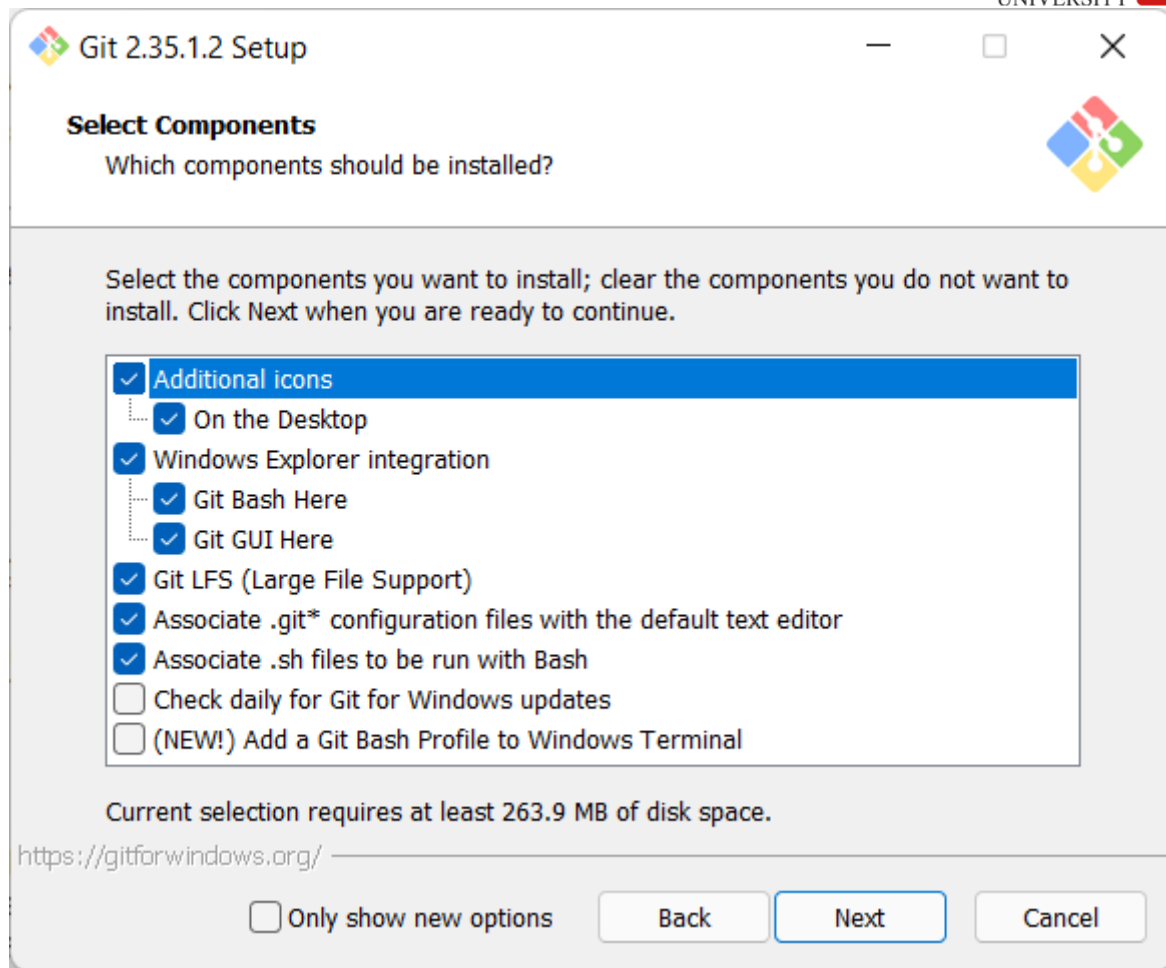
2. After opening this site, you have to select your operating system by clicking on it. Here I will show you the steps for the Windows operating system.



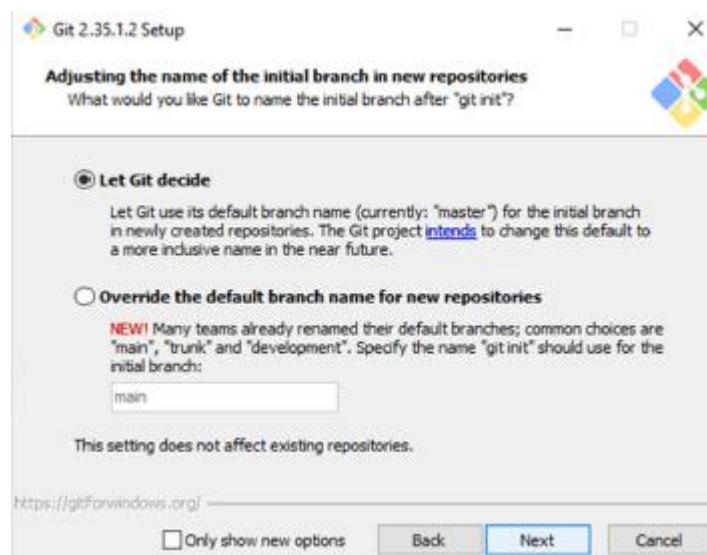
3. Now select the processor of the system you have. (Most of the system are now of 64-bits) After selecting the processor your download will start.
4. Now you have to open this folder.
5. After opening you will give a notification “Do you want to allow this app to make changes in your PC”
6. Click YES



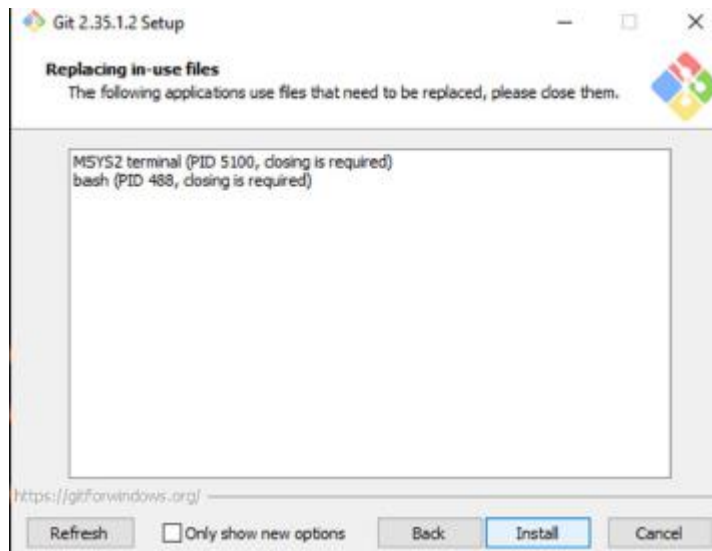
7. Click on Next



8. Continue clicking on next few times more



9. Now select the Install option.



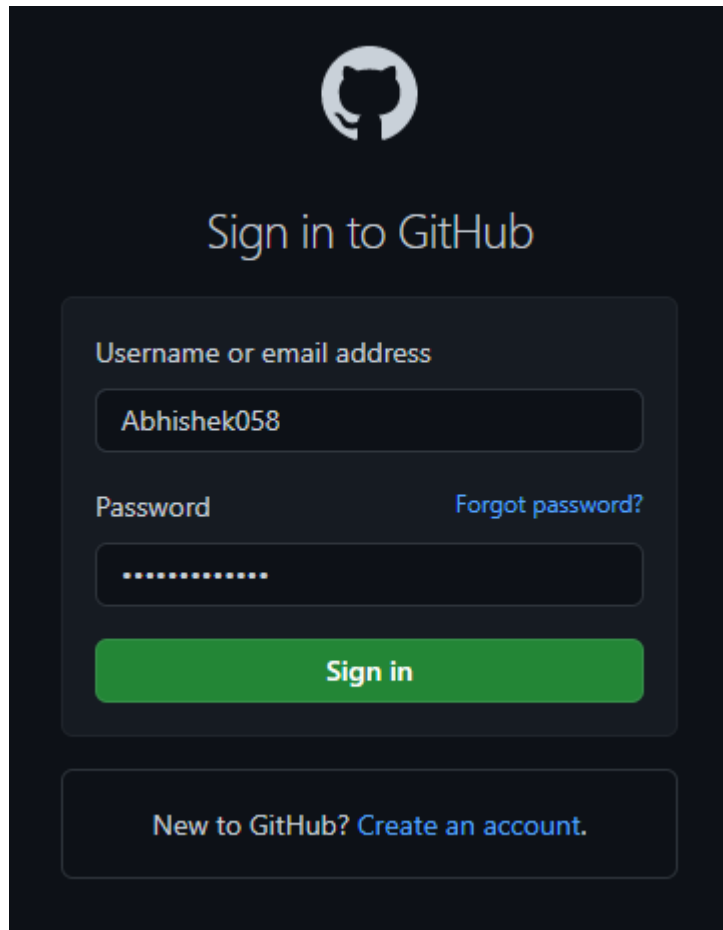
10. Click on Finish after the installation is finished.

11. The installation of the git is finished and now we have to setup git client and GitHub account.

EXPERIMENT NO:2

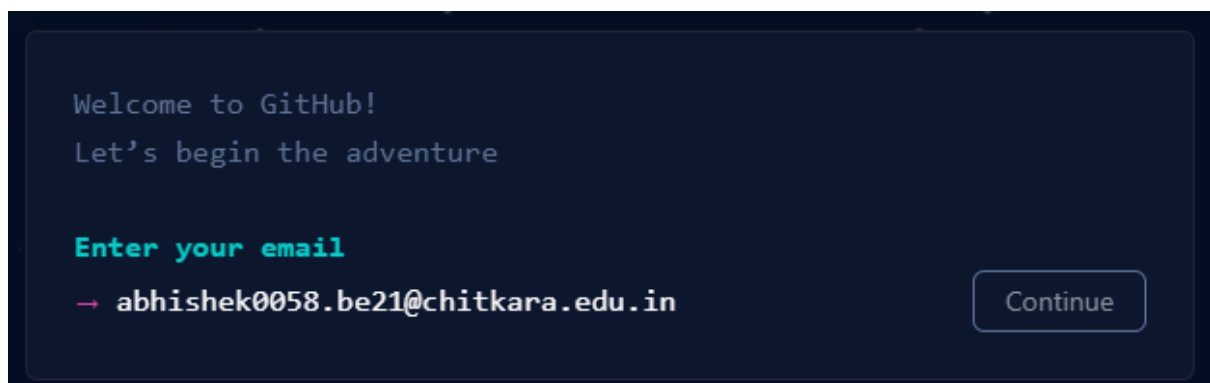
Aim: - Setting up GitHub Account

1. Open your web browser search GitHub login.
2. Click on Create an account if you are a new user or if you have already an account, please login.



The image shows the GitHub sign-in page. At the top is the GitHub Octocat logo. Below it, the text "Sign in to GitHub" is centered. There are two input fields: "Username or email address" with the value "Abhishek058" and "Password" with masked characters. A "Forgot password?" link is next to the password field. A green "Sign in" button is below the fields. At the bottom, a link says "New to GitHub? Create an account."

3. After Clicking on create a new account you will be redirected to a new page where you have to enter your email id which you want to use for your account. Now enter your password you want to create for your GitHub account. After that you will be asked to enter your username.



The image shows the GitHub welcome screen. It has a dark background with white text. The text reads: "Welcome to GitHub!", "Let's begin the adventure", and "Enter your email" in green. Below "Enter your email" is a red arrow pointing to the email address "abhishek0058.be21@chitkara.edu.in". A "Continue" button is on the right.

4. Now Click on Create Account.
5. Verify it from your email and you are all set to go.

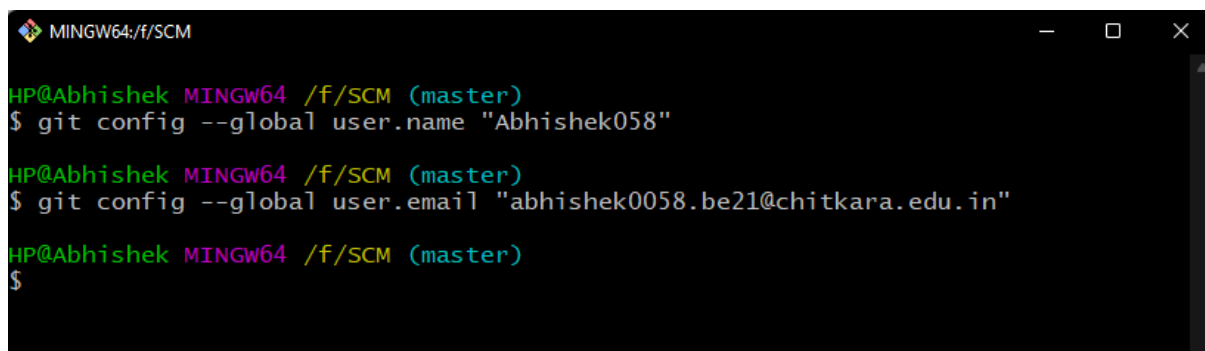
Aim: - Program to Generate logs

- When we use GIT for the first time, we have to give the user's name and email so that if I am going to change in project, it will be visible to all.

For this, we use command →

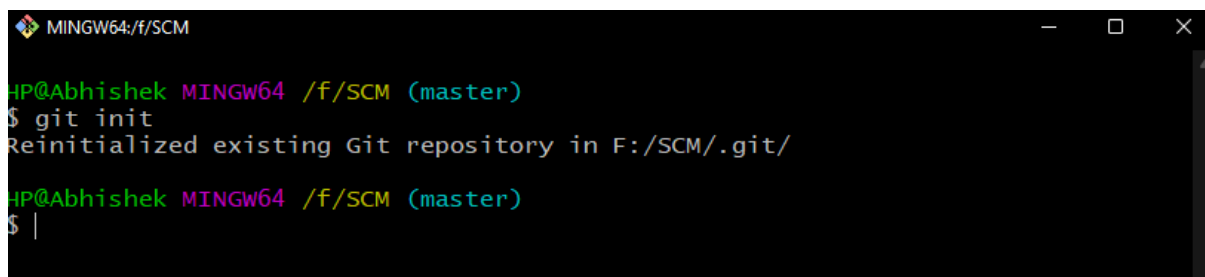
“Git config --global user.name “Name”

“Git config --global user. Email “email”



```
MINGW64:/f/SCM
HP@Abhishek MINGW64 /f/SCM (master)
$ git config --global user.name "Abhishek058"
HP@Abhishek MINGW64 /f/SCM (master)
$ git config --global user.email "abhishek0058.be21@chitkara.edu.in"
HP@Abhishek MINGW64 /f/SCM (master)
$
```

First of all, create a local repository using Git. For this, you have to make a folder in your device, right click and select “Git Bash Here”. This opens the Git terminal. To create a new local repository, use the command “git init” and it creates a folder. git.



```
MINGW64:/f/SCM
HP@Abhishek MINGW64 /f/SCM (master)
$ git init
Reinitialized existing Git repository in F:/SCM/.git/
HP@Abhishek MINGW64 /f/SCM (master)
$ |
```

Some Important Commands:

- **ls** → It gives the file names in the folder.
- **ls -l** → Gives the hidden files also.
- **git status** → Displays the state of the working directory and the staged snapshot.
- **touch filename** → This command creates a new file in the repository.

- **Clear** → It clears the terminal.
- **rm -rf. git** → It removes the repository.
- **git log** → displays all of the commits in a repository's history
- **git diff** → It compares my working tree to staging area.

Now, we have to create some files in the repository. Suppose we created index.html. Now type git status:

```

MINGW64:/f/SCM
HP@Abhishek MINGW64 /f/SCM (master)
$ ls
first.cpp  first.exe*

HP@Abhishek MINGW64 /f/SCM (master)
$ ls -lart
total 57
drwxr-xr-x 1 HP 197121      0 Mar 30 18:18 .vscode/
-rwxr-xr-x 1 HP 197121 44550 Mar 30 18:19 first.exe*
-rw-r--r-- 1 HP 197121   220 Apr  8 10:32 first.cpp
drwxr-xr-x 1 HP 197121      0 Apr  8 10:32 .git/
drwxr-xr-x 1 HP 197121      0 Apr  8 10:32 ./
drwxr-xr-x 1 HP 197121      0 Apr  8 19:13 ../

HP@Abhishek MINGW64 /f/SCM (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

HP@Abhishek MINGW64 /f/SCM (master)
$

```

You can see that index.html is in red colour that means it is an untracked file. Now firstly add the file in staging area and then commit the file.

For this, use command →

git add --a [For add all the files in staging area.]

git commit -m "write any message" [For commit the file]

- **git log:** The git log command displays a record of the commits in a Git repository. By default, the git log command displays a commit hash, the commit message, and other commit metadata.

```

MINGW64:/f/SCM
HP@Abhishek MINGW64 /f/SCM (master)
$ git add --a

HP@Abhishek MINGW64 /f/SCM (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    second.exe -> new.exe

HP@Abhishek MINGW64 /f/SCM (master)
$

```

```

MINGW64:/f/SCM
HP@Abhishek MINGW64 /f/SCM (master)
$ git log
commit 52c2230f9e9ca8a9d1ce6c053cc143177cb14492 (HEAD -> master, origin/master)
Author: Abhishek058 <abhishek0058.be21@chitkara.edu.in>
Date:   Mon Apr 11 23:03:58 2022 +0530

    updated

commit 1e3f1dd103262e6873a194aaffc6f0cec61aed64
Author: Abhishek058 <abhishek0058.be21@chitkara.edu.in>
Date:   Mon Apr 11 23:01:48 2022 +0530

    new commit

commit 145871091f39f9c1a2735429eae2e77041e51917
Author: Abhishek058 <abhishek0058.be21@chitkara.edu.in>
Date:   Mon Apr 11 22:57:12 2022 +0530

    updated

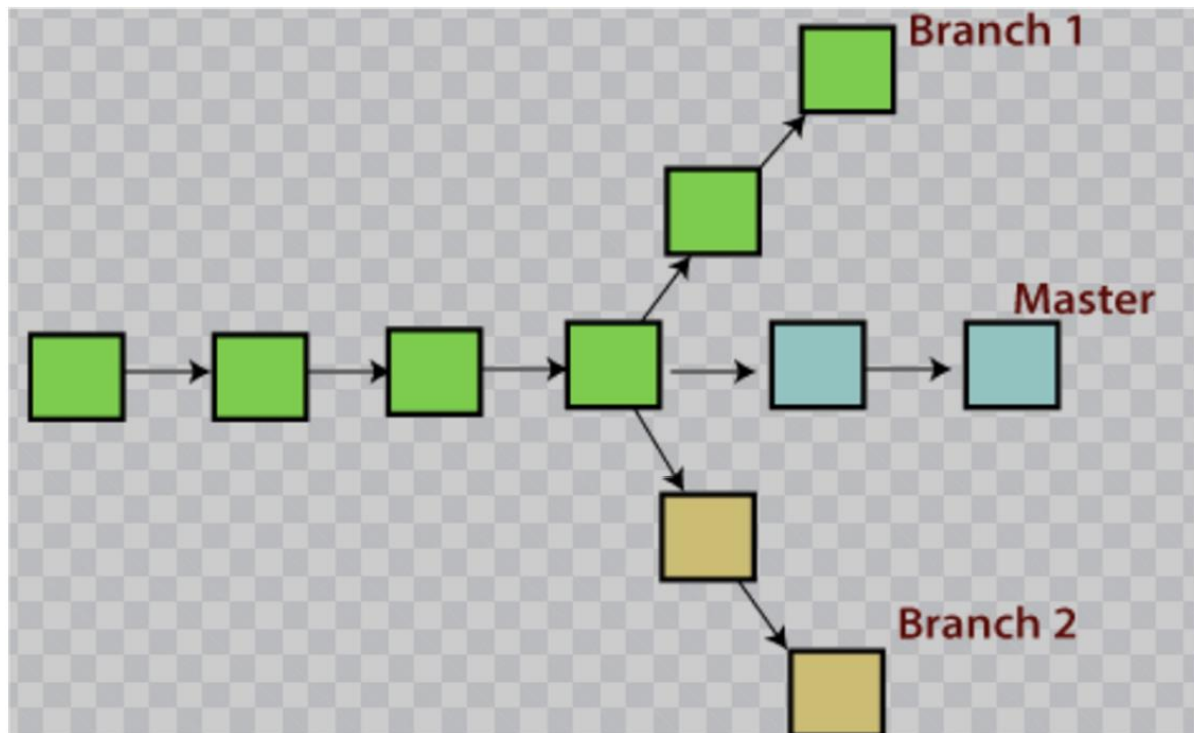
commit 5f58121831f8210bab42781101a14f077162737c
Author: Abhishek058 <abhishek0058.be21@chitkara.edu.in>
Date:   Mon Apr 11 22:48:42 2022 +0530

```

EXPERIMENT NO:4

Aim: - Create and visualize branches

A branch in Git is an independent line of work (a pointer to a specific commit). It allows users to create a branch from the original code (master branch) and isolate their work. Branches allow you to work on different parts of a project without impacting the main branch.



Let us see the command of it:

Firstly, add a new branch, let us suppose the branch name is activity1.

For this use command →

git branch name [adding new branch]

git branch [use to see the branch's names]

git checkout branch name [use to switch to the given branch]

```

MINGW64:/f/SCM
HP@Abhishek MINGW64 /f/SCM (master)
$ git branch branch2

HP@Abhishek MINGW64 /f/SCM (master)
$ checkout branch2
bash: checkout: command not found

HP@Abhishek MINGW64 /f/SCM (master)
$ git checkout branch2
Switched to branch 'branch2'

HP@Abhishek MINGW64 /f/SCM (branch2)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

HP@Abhishek MINGW64 /f/SCM (master)
$ git merge branch2
Already up to date.

HP@Abhishek MINGW64 /f/SCM (master)
$

```

As, we can see we can switch the branches by using “git checkout branch name” command. After, we are done with the branch that we have made to do master branch better we can merge it with the master branch.

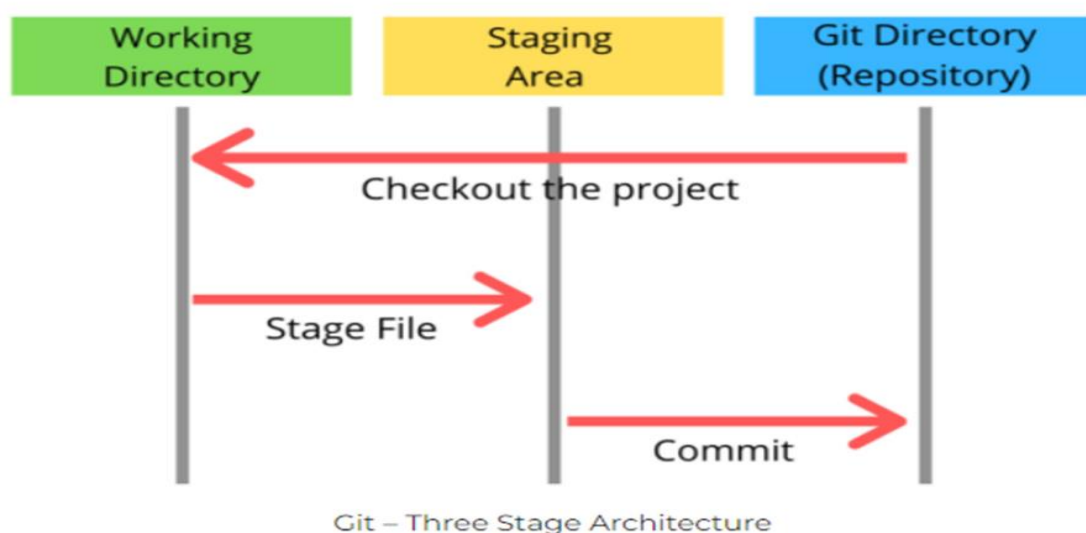
This can be done by using command: -

git merge branch name [that we have to add]

EXPERIMENT NO:5

Aim: - Git lifecycle description

. Now let's understand the three-stage architecture of Git



- **Working Directory:** This is the directory that we've initialized, and here all the changes are made to commit on GitHub.
- **Staging Area:** This is where we first put out code or files of the working repository. The command that we use to stage code is, "git add --a", "git add Filenames" or "git add -A". In simple terms, staging means telling Git what files we want to commit (new untracked files, modified files, or deleted files).
- **Git directory(repository):** This is where all the commits are stored whenever we make a commit. We can revert to an older version of our project using the **"Git checkout" command** from this directory.

```
HP@Abhishek MINGW64 /f/SCM (branch2)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

HP@Abhishek MINGW64 /f/SCM (master)
$ git merge branch2
Already up to date.
```

*Thank
you!*