SOURCE CODE MANAGEMENT FILE

Submitted by:

Name: Pratyukt Nag
Roll No. 2210990679
Group 27-A
Submission of:

Task 1.1
Task 1.2
Task 2

Submitted To:

Dr.Aditi Moudgil
Professor and Dean | Beta Cluster
Department of Computer Science & Engineering
Chitkara University Institute of Engineering and Technology
Rajpura, Punjab

Source Code Management File

Subject Name: **Source Code Management (SCM)**

Subject Code: **CS181**

Submitted By:

Name: Pratyukt Nag

Roll No. **2210990679**

Group: **G27-A**

Task 1.1 Submission (Week 4)

1. Setting up of Git Client
2. Setting up GitHub Account
3. Generate logs
4. Create and visualize branches
5. Git lifecycle description

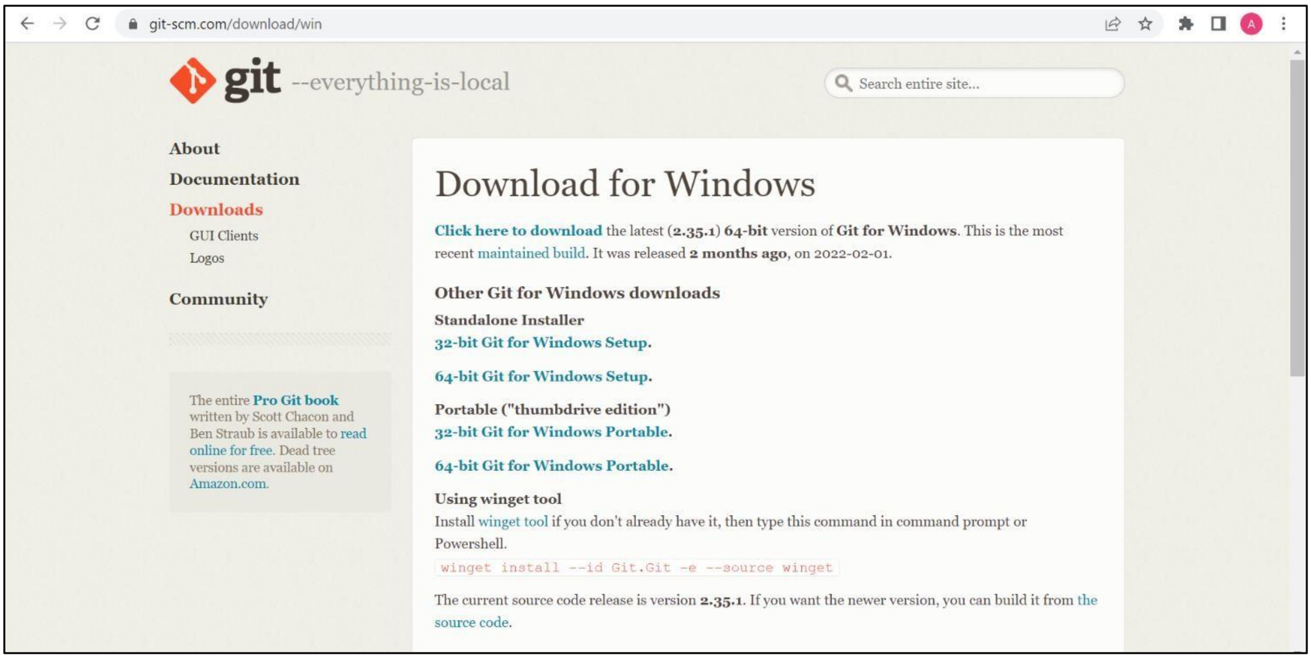
Experiment 1

Aim: Setting up of Git Client

Theory:
GIT: It’s a Version Control System (VCS). It is a software or we can say a server by which we are able to track all the previous changes in the code. It is basically used for pushing and pulling of code. We can use git and git-hub parallelly to work with multiple members or individually. We can make, edit, recreate, copy or download any code on git hub using git.

Procedure: We can install Git on Windows, using the most official build which is available for download on the GIT’s official website or by just typing (scm git) on any search engine. We can go on https://git-scm.com/download/win and can select the platform and bit-version to download. And after clicking on your desired bit-version or ios it will start downloading automatically.

Snapshots of download:



git-scm.com/download/win

git --everything-is-local

Search entire site...

About

Documentation

Downloads

- GUI Clients
- Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Download for Windows

Click here to download the latest (**2.35.1**) **64-bit** version of **Git for Windows**. This is the most recent maintained build. It was released **2 months ago**, on 2022-02-01.

Other Git for Windows downloads

Standalone Installer

32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Portable ("thumbdrive edition")

32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

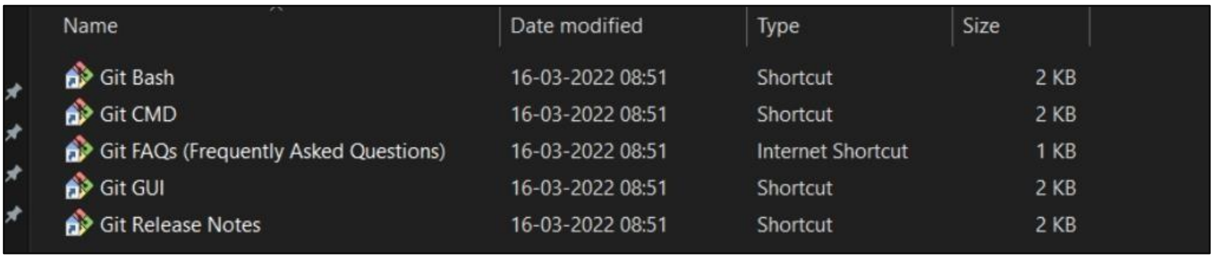
Using winget tool

Install winget tool if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

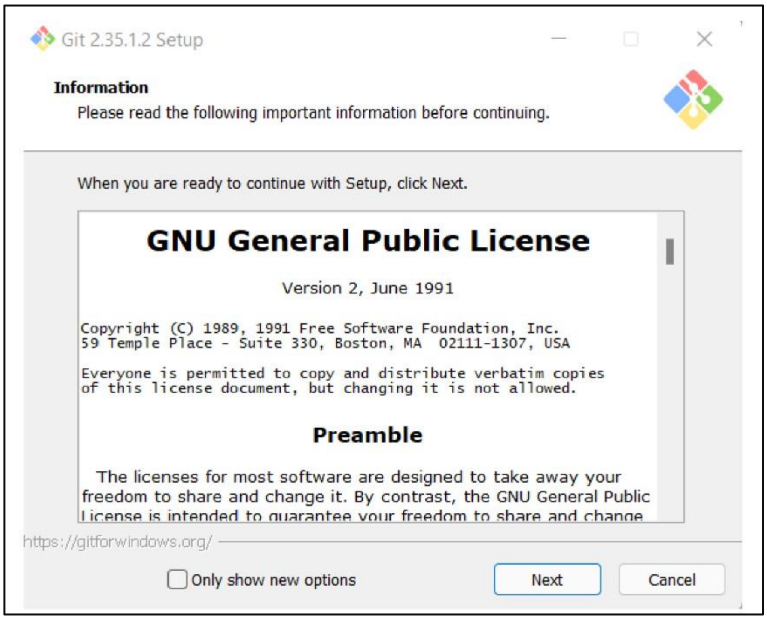
The current source code release is version **2.35.1**. If you want the newer version, you can build it from the source code.

Opted for “64-bit Git for Windows Setup”



Name	Date modified	Type	Size
Git Bash	16-03-2022 08:51	Shortcut	2 KB
Git CMD	16-03-2022 08:51	Shortcut	2 KB
Git FAQs (Frequently Asked Questions)	16-03-2022 08:51	Internet Shortcut	1 KB
Git GUI	16-03-2022 08:51	Shortcut	2 KB
Git Release Notes	16-03-2022 08:51	Shortcut	2 KB

Git and its files in downloads



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

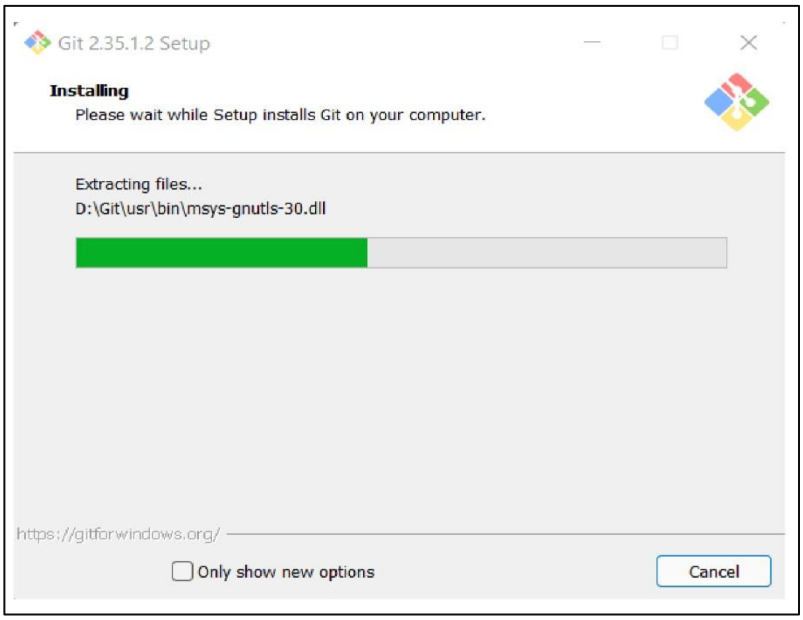
Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

Git Setup

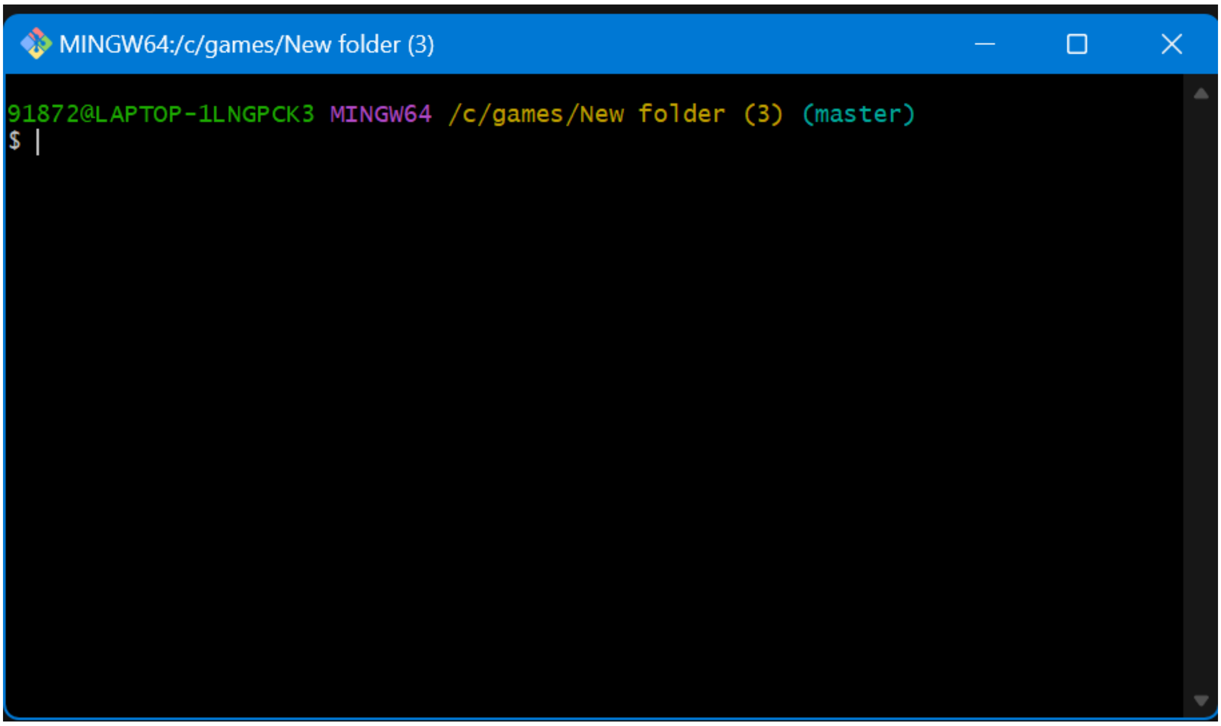


Installing

Please wait while Setup installs Git on your computer.

Extracting files...
D:\Git\usr\bin\msys-gnutls-30.dll

Git Installation



```
91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/New folder (3) (master)
$ |
```

Git Bash Launched

Experiment 2

Aim: Setting up GitHub Account

Theory:

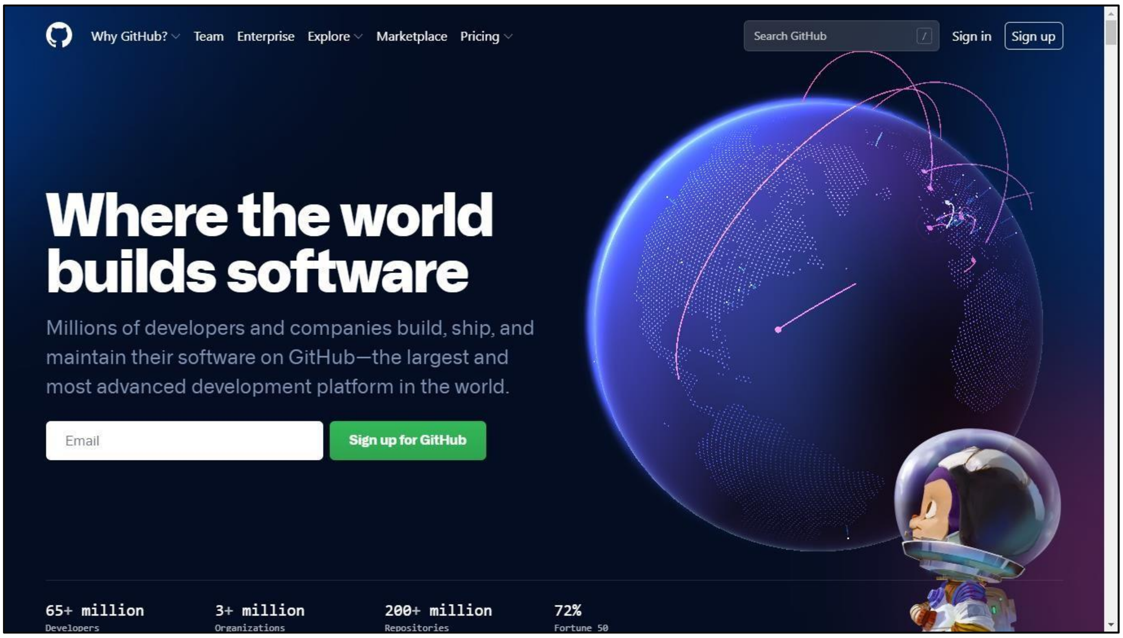
GitHub: GitHub is a website and cloud-based service (client) that helps an individual or developers to store and manage their code. We can also track as well as control changes to our or public code.

Advantages of GitHub: GitHub has a user-friendly interface and is easy to use.We can connect the git-hub and git but using some commands shown below in figure 001. Without GitHub we cannot use Git because it generally requires a host and if we are working for a project, we need to share it will our team members, which can only be done by making a repository. Additionally, anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

Procedure:

To make an account on GitHub, we search for GitHub on our browser or visit https://github.com/signup. Then, we will enter our mail ID and create a username and password for a GitHub account.

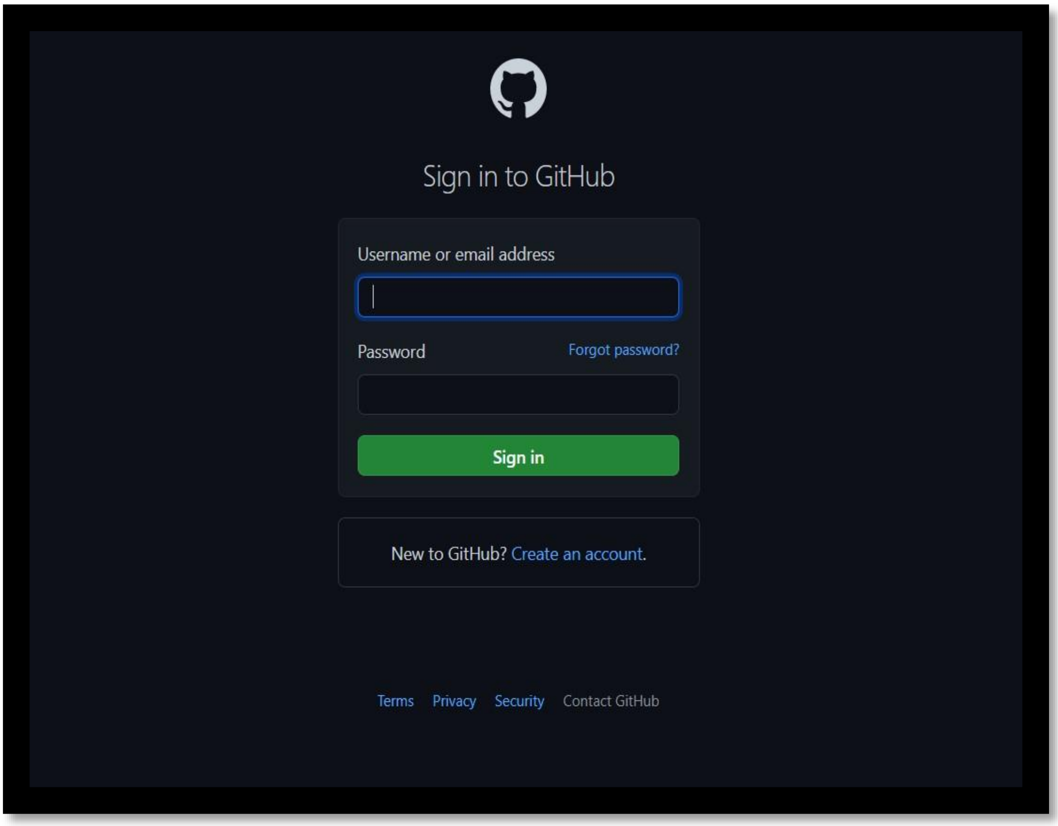
Snapshots _



Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

After visiting the link this type of interface will appear, if you already have an account, you can sign in and if not, you can create.



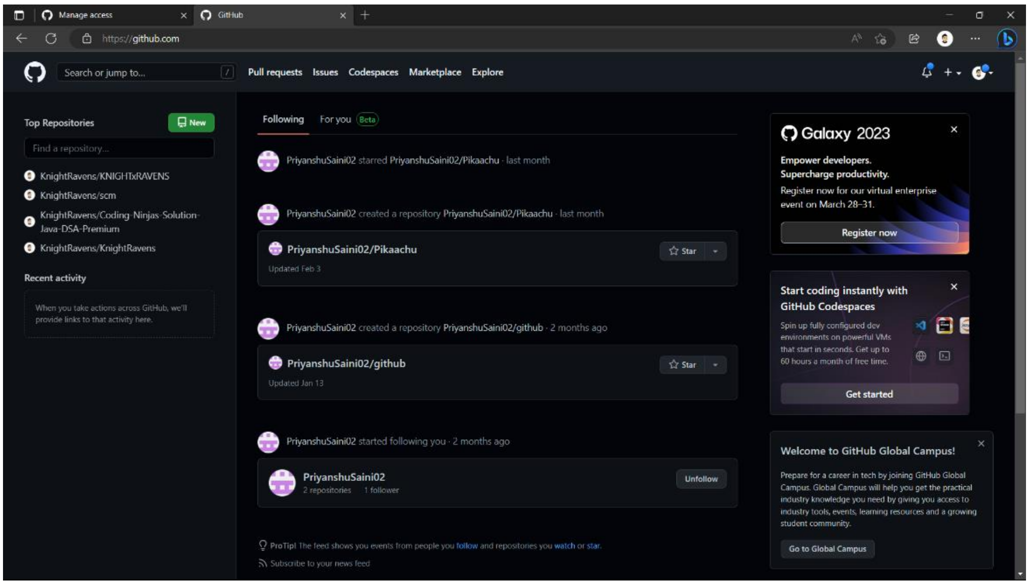
Sign in to GitHub

Username or email address

Password

New to GitHub? Create an account.

GitHub Login



KnightRavens/KNIGHTxRAVENS

KnightRavens/scm

KnightRavens/Coding-Ninjas-Solution-Java-DSA-Premium

KnightRavens/KnightRavens

When you take actions across GitHub, we'll provide links to that activity here.

PriyanshuSaini02 starred PriyanshuSaini02/Pikaachu · last month

PriyanshuSaini02 created a repository PriyanshuSaini02/Pikaachu · last month

PriyanshuSaini02/Pikaachu
Updated Feb 3

PriyanshuSaini02 created a repository PriyanshuSaini02/github · 2 months ago

PriyanshuSaini02/github
Updated Jan 13

PriyanshuSaini02 started following you · 2 months ago

PriyanshuSaini02
2 repositories 1 follower

ProTip! The feed shows you events from people you follow and repositories you watch or star.

Subscribe to your news feed

Galaxy 2023

Empower developers.
Supercharge productivity.

Register now for our virtual enterprise event on March 28–31.

Start coding instantly with GitHub Codespaces

Spin up fully configured dev environments on powerful VMs that start in seconds. Get up to 60 hours a month of free time.

Welcome to GitHub Global Campus!

Prepare for a career in tech by joining GitHub Global Campus. Global Campus will help you get the practical industry knowledge you need by giving you access to industry tools, events, learning resources and a growing student community.

GitHub Interface

Experiment 3

Aim: Program to Generate log

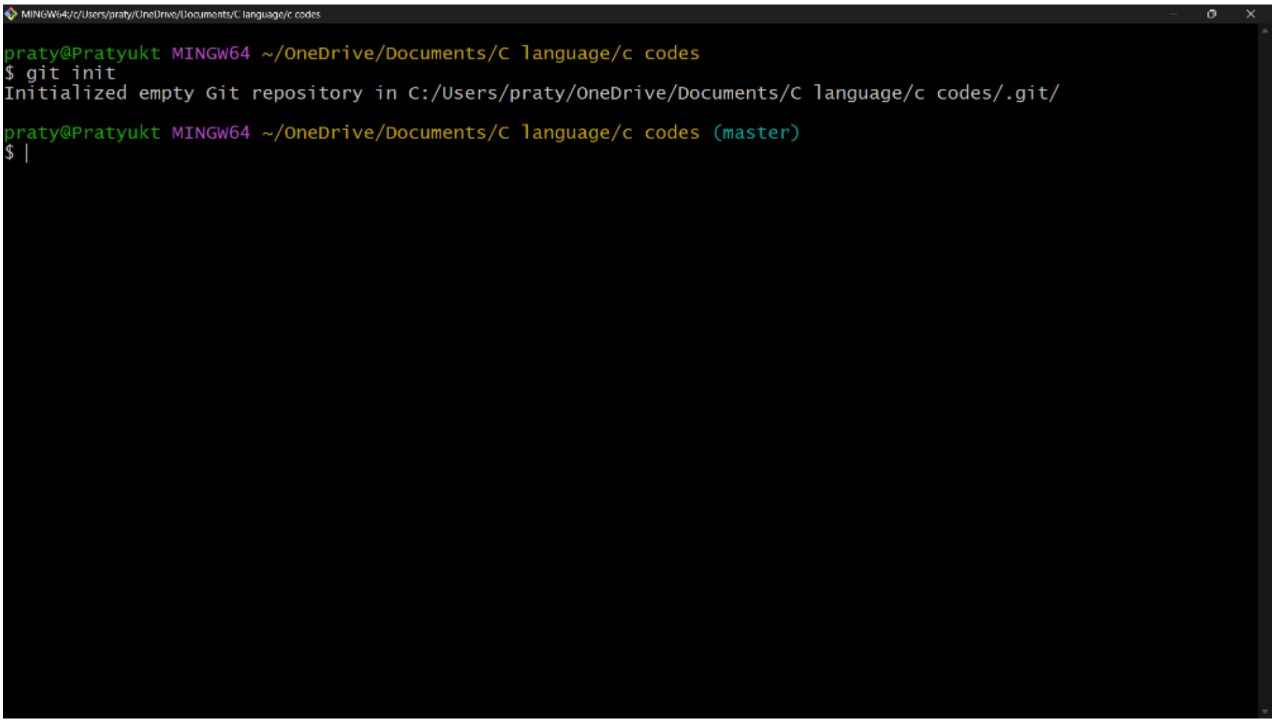
Theory:

Logs: Logs are nothing but the history which we can see in git by using the code git log.

It contains all the past commits, insertions and deletions in it which we can see any time. Logs helps to check that what were the changes in the code or any other file and by whom. It also contains the number of insertions and deletions including at which time it was changed.

Procedure:

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:/c/Users/praty/OneDrive/Documents/C language/c codes

```
praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes
$ git init
Initialized empty Git repository in C:/Users/praty/OneDrive/Documents/C language/c codes/.git/

praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes (master)
$ |
```

When we use GIT for the first time, we have to give the user 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”

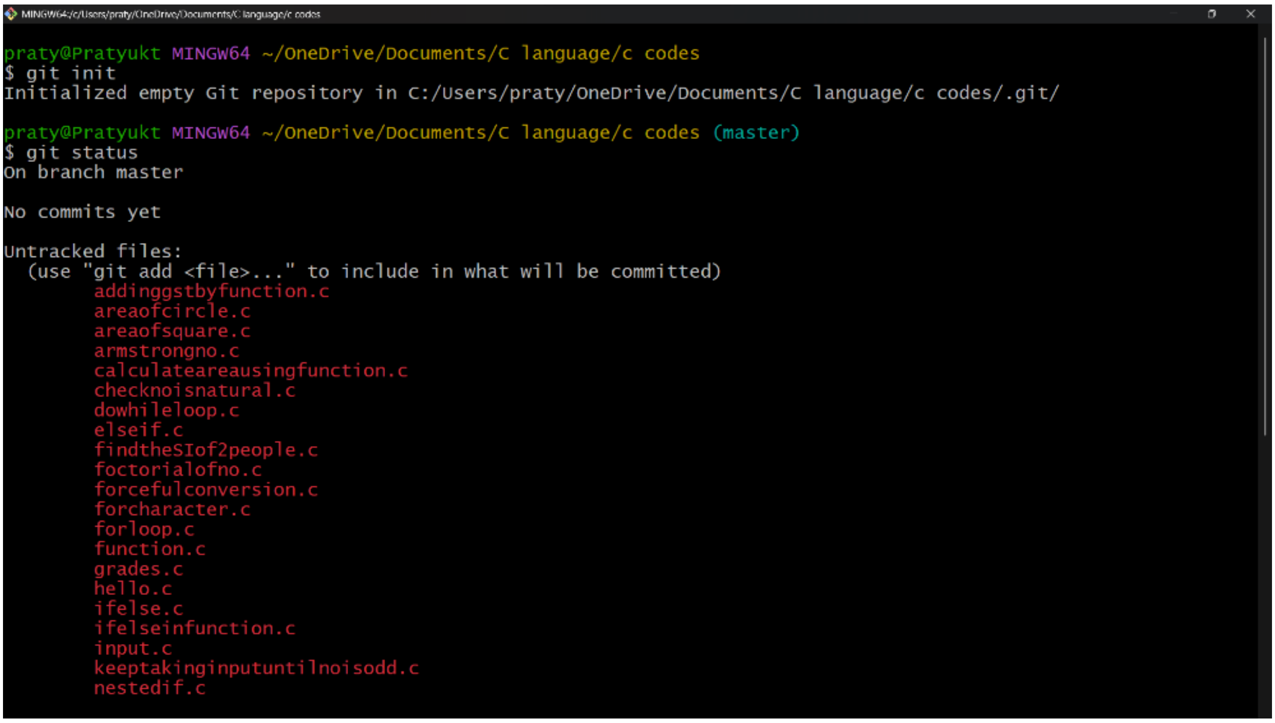
For verifying the user’s name and email, we use:

“git config --global user.name”
“git config --global user.email”

Some Important Commands

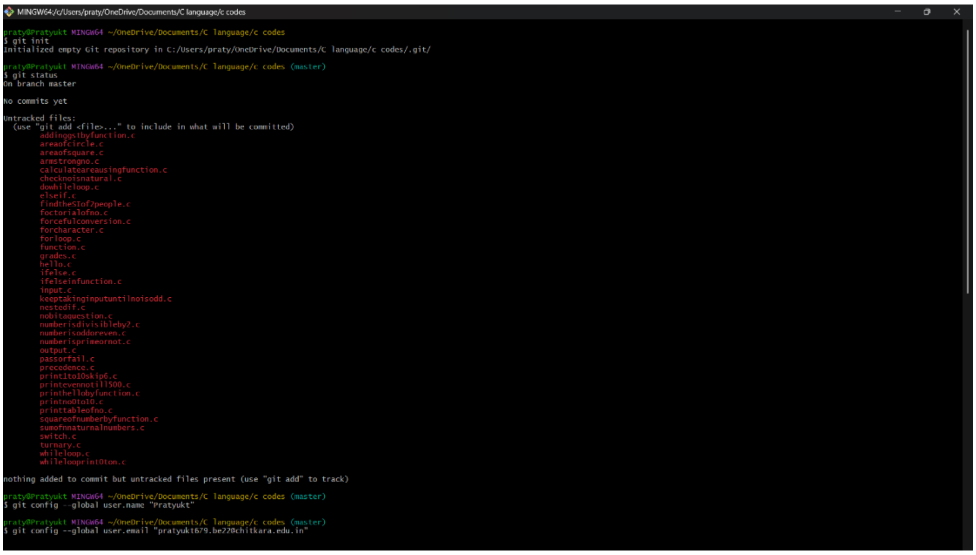
- ls ❼ It gives the file names in the folder.
- ls -lart ❼ Gives the hidden files also.
- git status ❼ Displays the state of the working directory and the staged snapshot.
- touch filename ❼ This command creates a new file in the repository.
- Clear ❼ It clears the terminal.
- rm -rf .git ❼ It removes the repository.
- git log ❼ displays all of the commits in a repository's history
- git diff ❼ It compares my working tree to staging area.

Git Status:



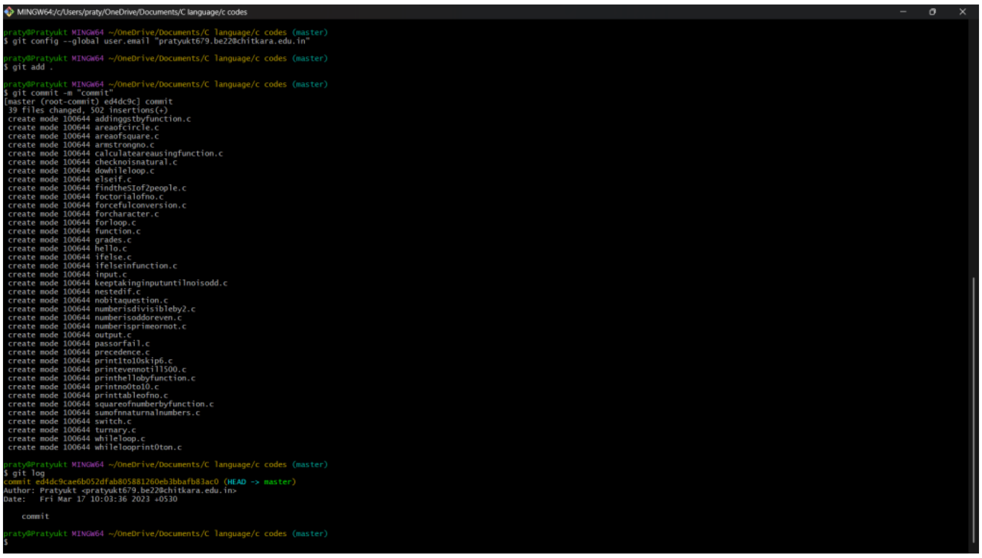
```
MINGW64:/c/Users/praty/OneDrive/Documents/C language/c codes

praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes
$ git init
Initialized empty Git repository in C:/Users/praty/OneDrive/Documents/C language/c codes/.git/

praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        addinggstbyfunction.c
        areaofcircle.c
        areaofsquare.c
        armstrongno.c
        calculateareausingfunction.c
        checknoisnatural.c
        dowhileloop.c
        elseif.c
        findtheSIof2people.c
        foctorialofno.c
        forcefulconversion.c
        forcharacter.c
        forloop.c
        function.c
        grades.c
        hello.c
        ifelse.c
        ifelseinfunction.c
        input.c
        keeptakinginputuntilnoisodd.c
        nestedif.c
```

Git Log:



MINGW64:/c/Users/praty/OneDrive/Documents/C language/c codes

```
praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes
$ git init
Initialized empty Git repository in C:/Users/praty/OneDrive/Documents/C language/c codes/.git/

praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        addinggstbyfunction.c
        areaofcircle.c
        areaofsquare.c
        armstrongno.c
        calculateareausingfunction.c
        checknoisnatural.c
        dowhileloop.c
        elseif.c
        findtheSIof2people.c
        foctorialofno.c
        forcefulconversion.c
        forcharacter.c
        forloop.c
        function.c
        grades.c
        hello.c
        ifelse.c
        ifelseinfunction.c
        input.c
        keeptakinginputuntilnoisodd.c
        nestedif.c
        nobitaquestion.c
        numberisdivisibleby2.c
        numberisoddoreven.c
        numberisprimeornot.c
        output.c
        passorfail.c
        precedence.c
        print1to10skip6.c
        printevennotill500.c
        printhellobyfunction.c
        printno0to10.c
        printtableofno.c
        squareofnumberbyfunction.c
        sumofnnaturnalnumbers.c
        switch.c
        turnary.c
        whileloop.c
        whileloopprint0ton.c

nothing added to commit but untracked files present (use "git add" to track)

praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes (master)
$ git config --global user.name "Pratyukt"

praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes (master)
$ git config --global user.email "pratyukt679.be22@chitkara.edu.in"
```



MINGW64:/c/Users/praty/OneDrive/Documents/C language/c codes

```
praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes (master)
$ git config --global user.email "pratyukt679.be22@chitkara.edu.in"

praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes (master)
$ git add .

praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes (master)
$ git commit -m "commit"
[master (root-commit) ed4dc9c] commit
 39 files changed, 502 insertions(+)
 create mode 100644 addinggstbyfunction.c
 create mode 100644 areaofcircle.c
 create mode 100644 areaofsquare.c
 create mode 100644 armstrongno.c
 create mode 100644 calculateareausingfunction.c
 create mode 100644 checknoisnatural.c
 create mode 100644 dowhileloop.c
 create mode 100644 elseif.c
 create mode 100644 findtheSIof2people.c
 create mode 100644 foctorialofno.c
 create mode 100644 forcefulconversion.c
 create mode 100644 forcharacter.c
 create mode 100644 forloop.c
 create mode 100644 function.c
 create mode 100644 grades.c
 create mode 100644 hello.c
 create mode 100644 ifelse.c
 create mode 100644 ifelseinfunction.c
 create mode 100644 input.c
 create mode 100644 keeptakinginputuntilnoisodd.c
 create mode 100644 nestedif.c
 create mode 100644 nobitaquestion.c
 create mode 100644 numberisdivisibleby2.c
 create mode 100644 numberisoddoreven.c
 create mode 100644 numberisprimeornot.c
 create mode 100644 output.c
 create mode 100644 passorfail.c
 create mode 100644 precedence.c
 create mode 100644 print1to10skip6.c
 create mode 100644 printevennotill500.c
 create mode 100644 printhellobyfunction.c
 create mode 100644 printno0to10.c
 create mode 100644 printtableofno.c
 create mode 100644 squareofnumberbyfunction.c
 create mode 100644 sumofnnaturnalnumbers.c
 create mode 100644 switch.c
 create mode 100644 turnary.c
 create mode 100644 whileloop.c
 create mode 100644 whileloopprint0ton.c

praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes (master)
$ git log
commit ed4dc9cae6b052dfab805881260eb3bbafb83ac0 (HEAD -> master)
Author: Pratyukt <pratyukt679.be22@chitkara.edu.in>
Date:   Fri Mar 17 10:03:36 2023 +0530

    commit

praty@Pratyukt MINGW64 ~/OneDrive/Documents/C language/c codes (master)
$
```

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.

Experiment 4

Aim: Create and visualize branches

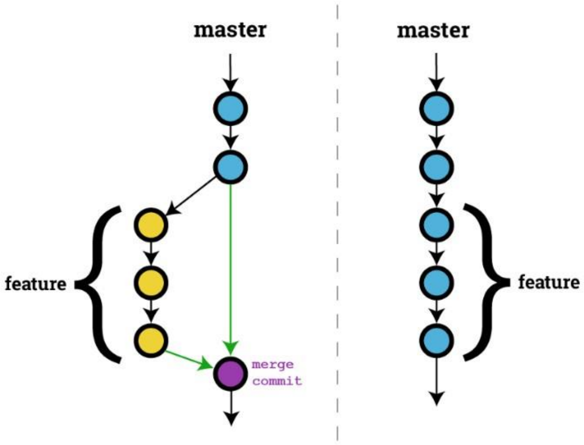
Theory:

Branching: 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.

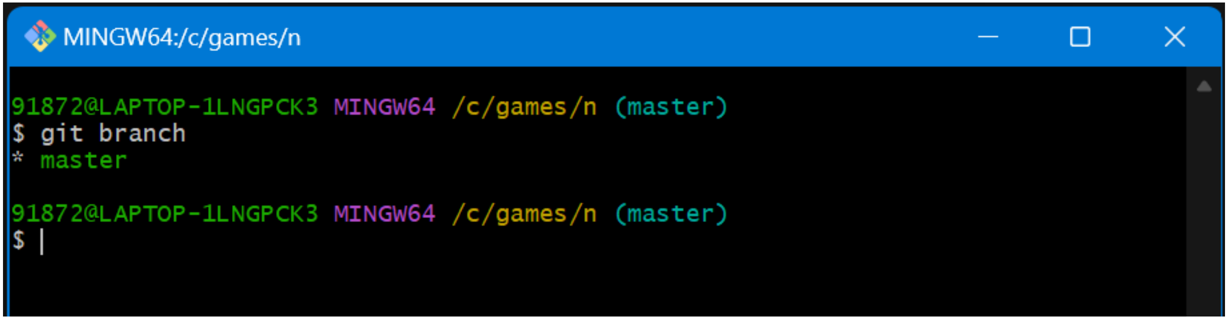
Create branches: The main branch in git is called as master branch. But we can make branches out of this main master branch. All the files present in master can be shown in branch but the file which are created in branch are not shown in master branch. We can also merge both the parent (master) and child (other branches).

Syntax:

For creating a new branch, git branch name by default is master branch.

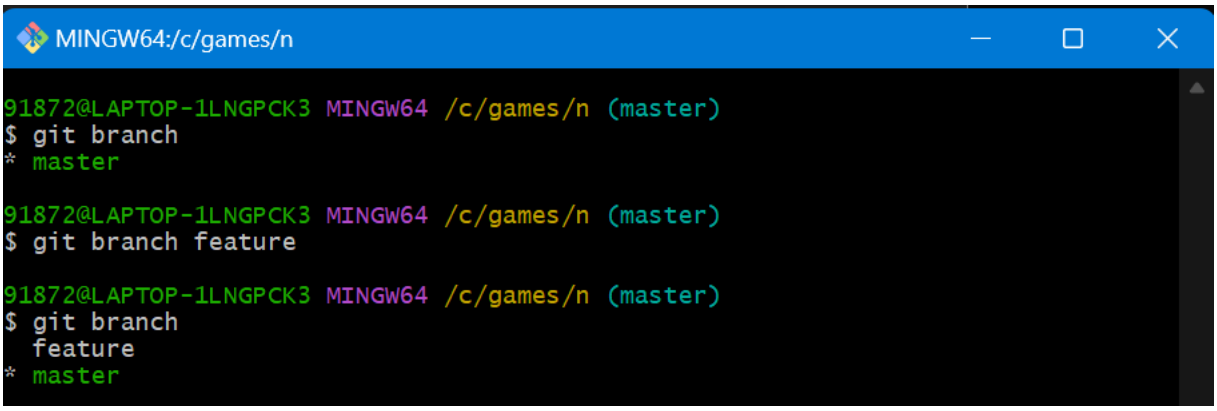


Snapshots –



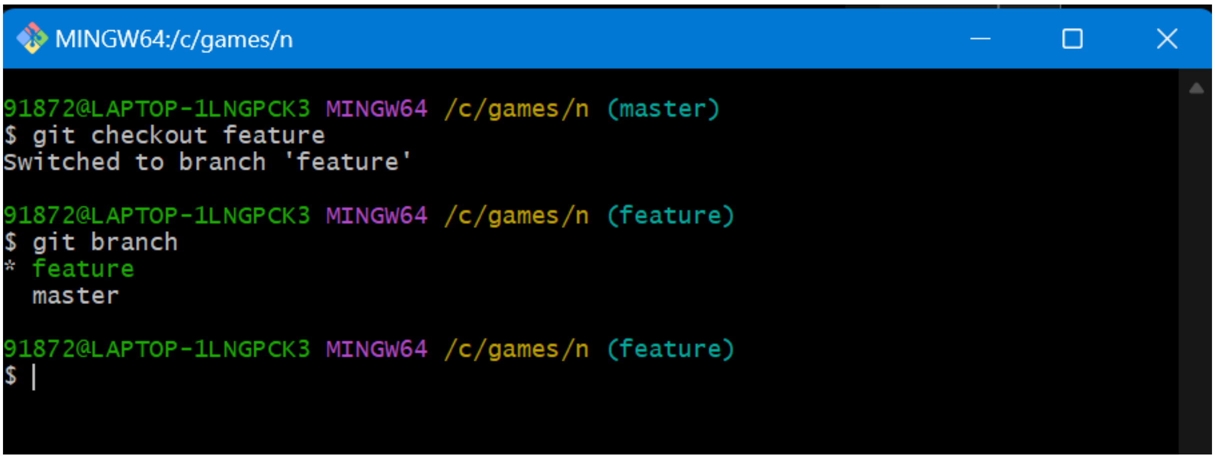
```
91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/n (master)
$ git branch
* master

91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/n (master)
$ |
```

Default branch is master branch



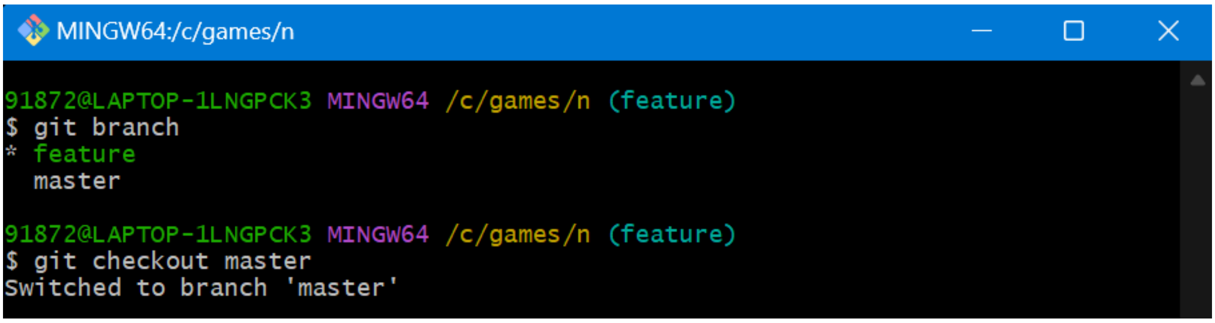
```
91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/n (master)
$ git branch
* master

91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/n (master)
$ git branch feature

91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/n (master)
$ git branch
  feature
* master
```

Adding a feature branch



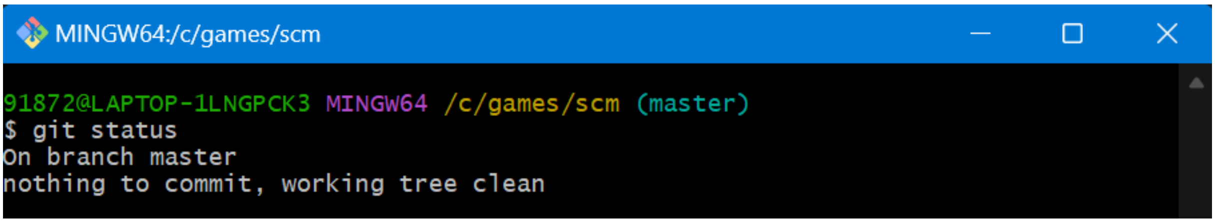
```
91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/n (master)
$ git checkout feature
Switched to branch 'feature'

91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/n (feature)
$ git branch
* feature
  master

91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/n (feature)
$ |
```

Switching branch



```
91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/n (feature)
$ git branch
* feature
  master

91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/n (feature)
$ git checkout master
Switched to branch 'master'
```

Switching to master branch



```
91872@LAPTOP-1LNGPCK3 MINGW64 /c/games/scm (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Checking commits

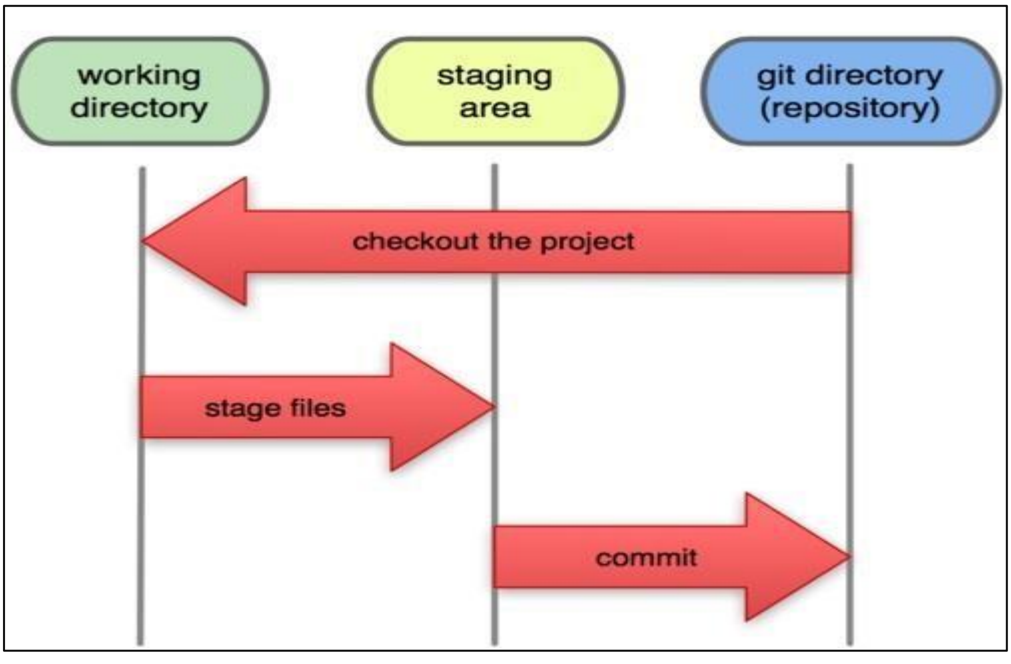
Experiment 5

Aim: Git lifecycle description

Theory:

Stages in GIT Life Cycle: Files in a Git project have various stages like Creation, Modification, Refactoring, and Deletion and so on. Irrespective of whether this project is tracked by Git or not, these phases are still prevalent. However, when a project is under Git version control system, they are present in three major Git states in addition to these basic ones. Here are the three Git states:

- Working directory
- Staging area
- Git directory



Working Directory:

Consider a project residing in your local system. This project may or may not be tracked by Git. In either case, this project directory is called your Working directory.

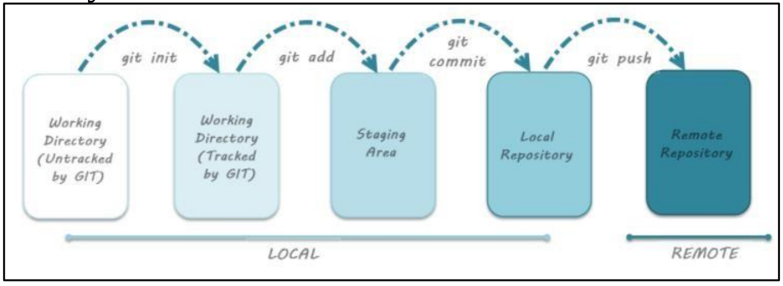
Staging Area:

Staging area is the playground where you group, add and organize the files to be committed to Git for tracking their versions.

Git Directory:

Now that the files to be committed are grouped and ready in the staging area, we can commit these files. So, we commit this group of files along with a commit message explaining what is the commit about. Apart from commit message, this step also records the author and time of the commit. Now, a snapshot of the files in the commit is recorded by Git. The information related to this commit is stored in the Git directory.

Remote Repository: It means mirror or clone of the local Git repository in GitHub. And pushing means uploading the commits from local Git repository to remote repository hosted in GitHub.


git init
git add
git commit
git push
Working Directory (Untracked by GIT)
Working Directory (Tracked by GIT)
Staging Area
Local Repository
Remote Repository
LOCAL
REMOTE