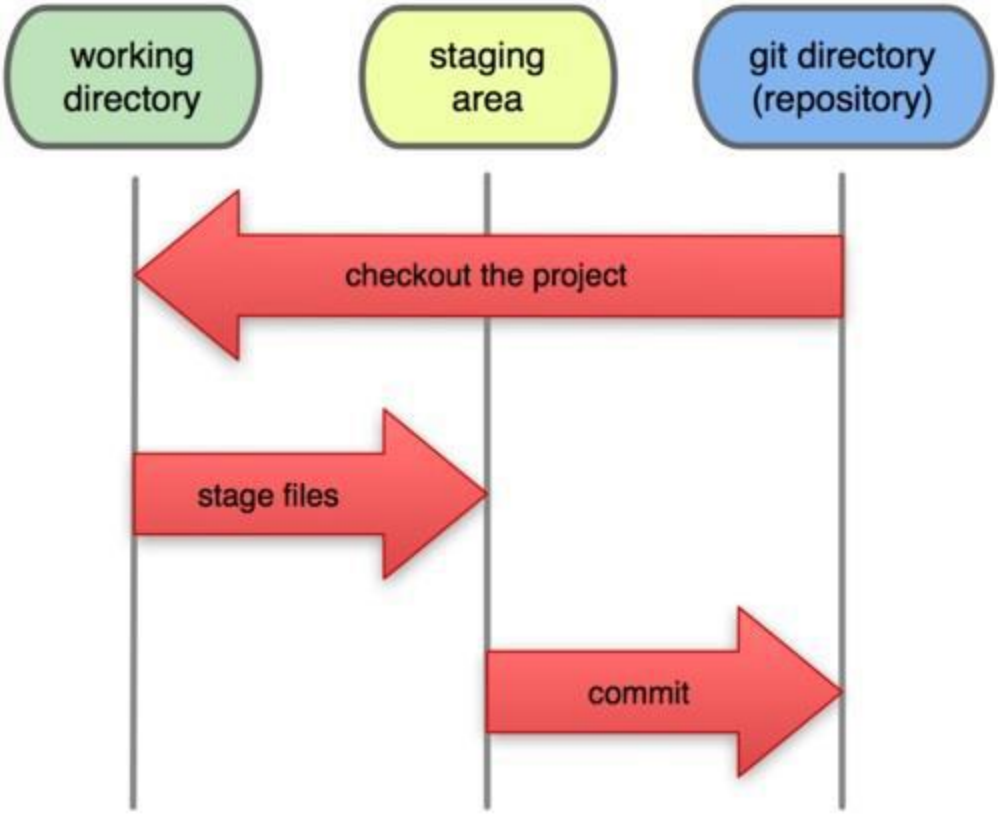
Experiment – 5

Aim: Git Life Cycle 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 File Name" 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 or project using the "git checkout" command from this directory.

Some Important Commands

- **Git operations and commands:**

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**.

- 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"

- **ls** → It gives the file names in the folder.
- **ls -lart** → 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.