Subject Name: **Source Code Management**

Subject Code: **22CS003**

Session: **2022-23**

Department: **DCSE**

****

|  |  |  |
| --- | --- | --- |
| **Submitted By:**  Muskan Aggarwal  2210991946  G11-A |  | **Submitted To:**  Ridhima mam |

|  |  |  |
| --- | --- | --- |
| **S. No** | **Program Title** | **Page No.** |
| 1 | Setting up of Git Client |  |
| 2 | Setting up GitHub Account |  |
| 3 | Git life cycle description |  |
| 4 | Add collaboration on git hub repo |  |
| 5 | Create and visualise branches |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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.

**Steps:**

1)Go to Git scm.com



2)Click Download for Windows



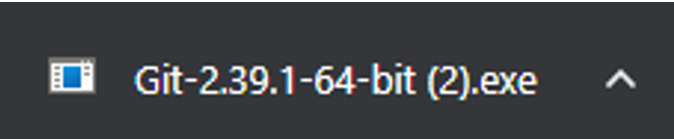
3)Click on 64 bit for Window Setup

1)standalone installer

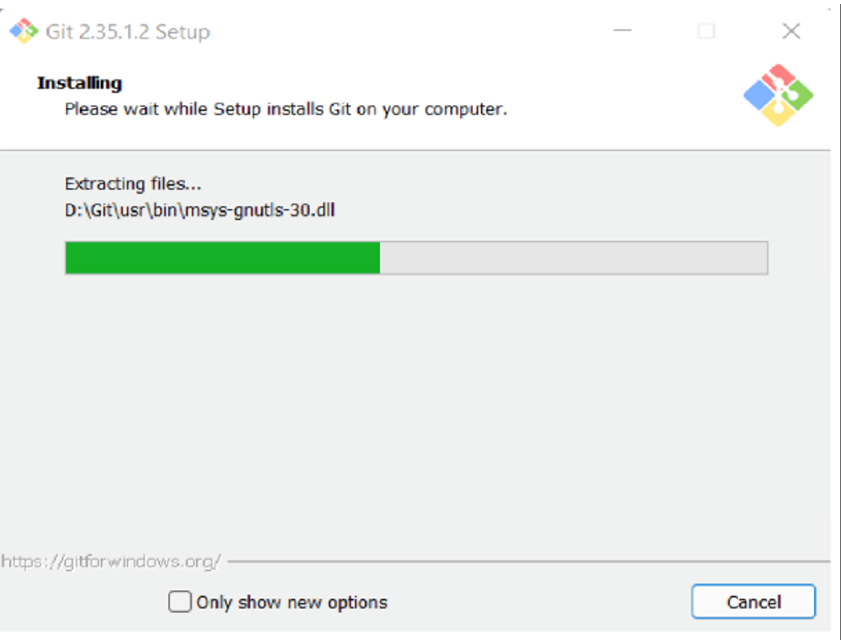
32 bit , 64 bit

2)portable (“thumb drive edition”)

32 bit , 64 bit4) run.exe file



5)INSTALLING GIT

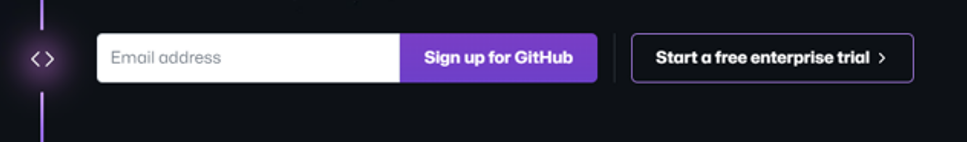


**Experiment 2 Aim: Setting up GitHub Account**

**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.

**Steps :**

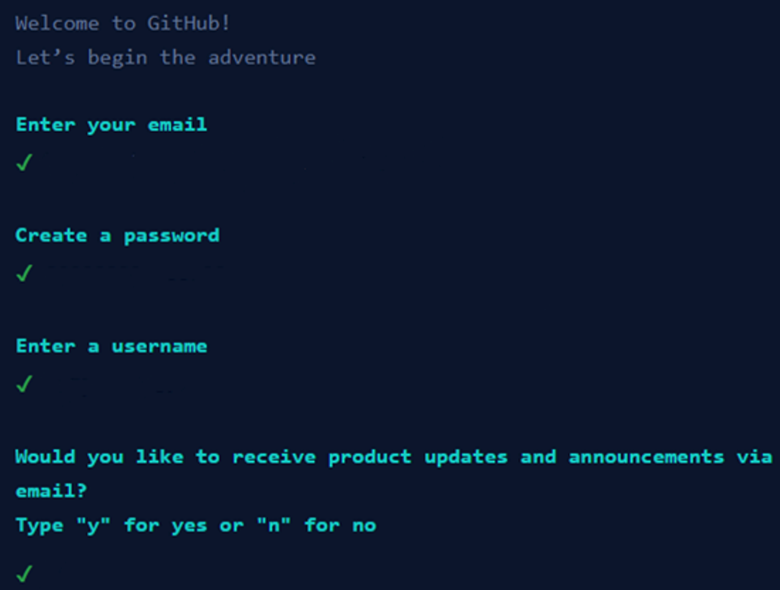
1)Go to hit hub.com



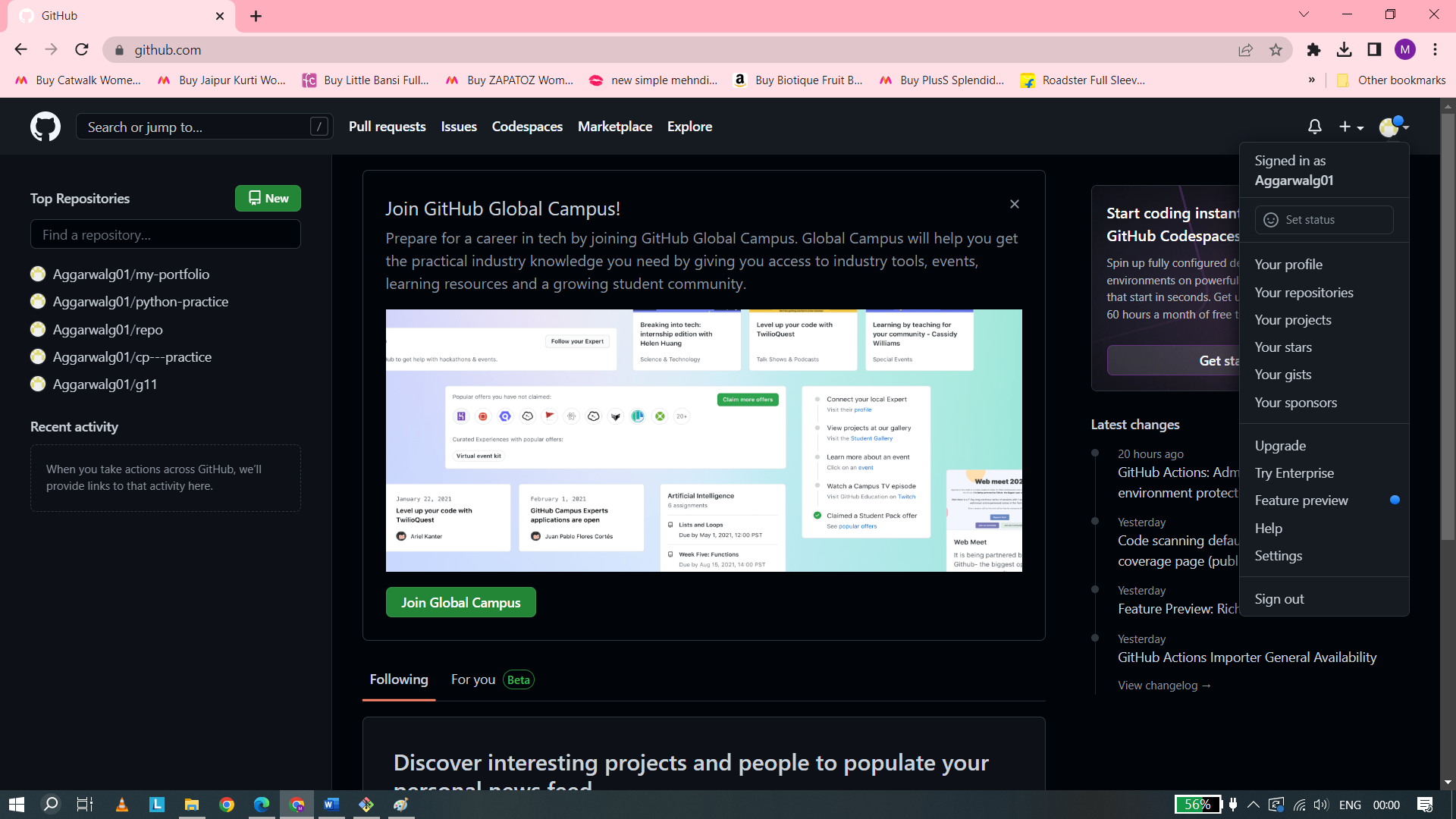
2)Enter your email address

3)Create a password

4)Setting up your account

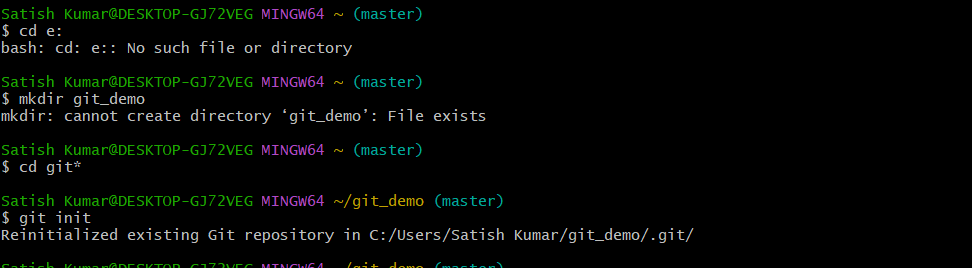


5)Your Account Has Been Setup



**Experiment 3 Aim: Program to Generate log**

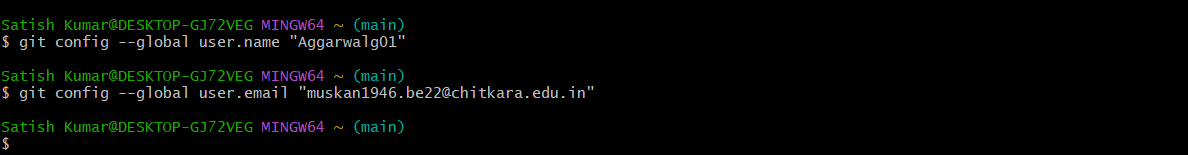
**Theory**: Logs: Logs are nothing but the history which we can see in git by using the code git log. It contains all the past commits, insertions and deletions in it which we can see any time. Logs helps to check that what were the changes in the code or any other file and by whom. It also contains 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”.



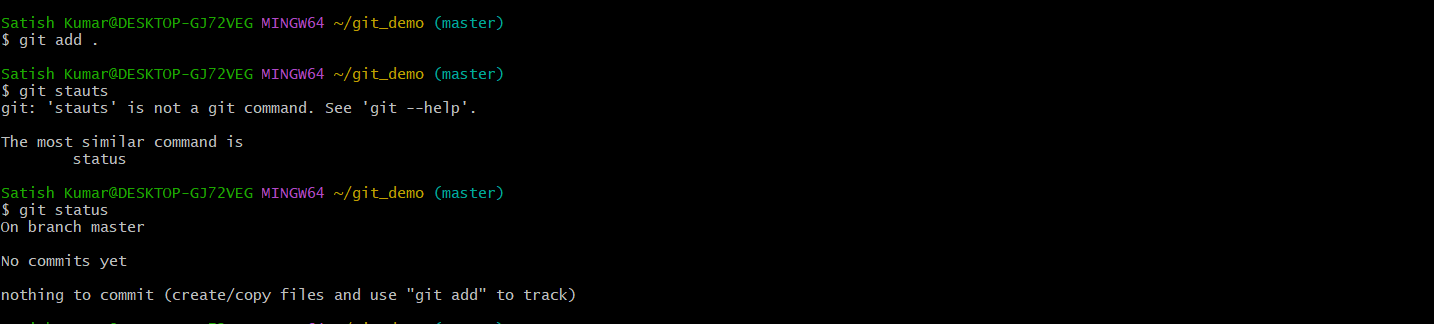
1. **Cd C :** It is used select a drive so that we can store a repository in that drive . here we select C drive.
2. **Mkdir git\_demo :** It is used to make a directory . we give the name git\_demo to our directory.
3. **Cd git\* :** It is used to call the present directory which we are

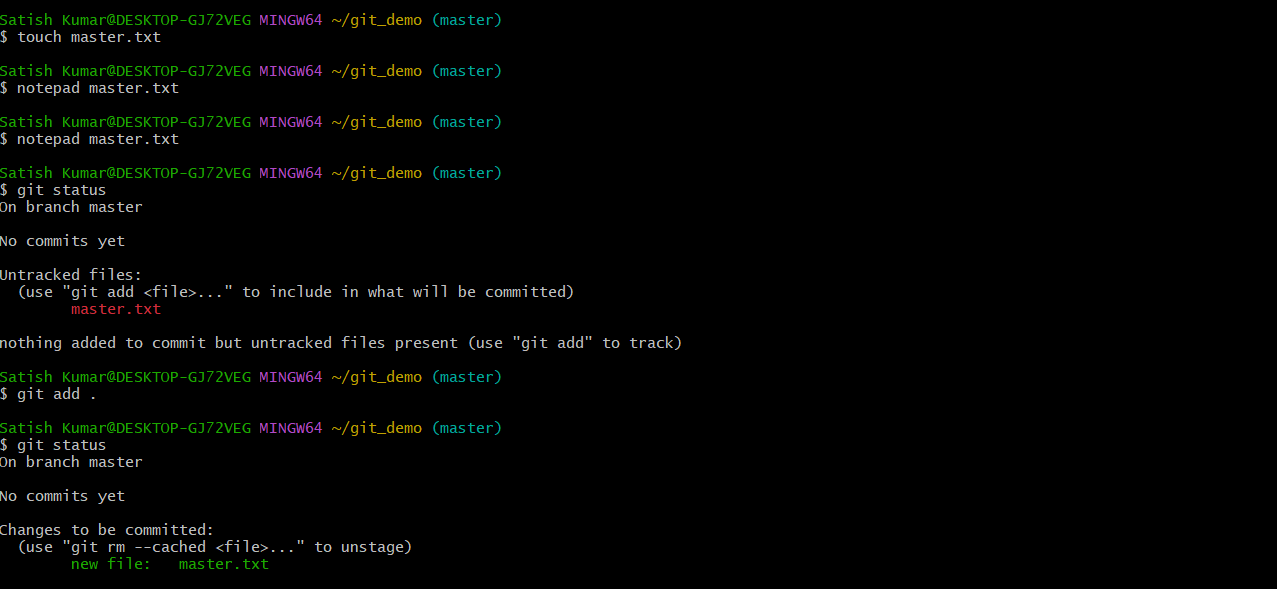
using at present.

1. **Git init :** It is used to initialise the empty git repository .
2. **Pwd :** we can use use this command to see that which directory we are working presently



1)**git config :** It is tell us about all the command that we can use during further process





1)**git config –global user.name “name”:** it allow to set a global project username.

Name is used that we have given during creating the github account.

I used “Aggarwalg01”

2)**git config –global user.email “email adress”:** it allow to set a global project email address. Email.address is used that address we have given during creating github account.

Mine is “muskan1946.be22@chitkara.edu.in”

3)**touch name.txt :** it is used when we have a directory on my desktop created using powershell and now we are trying to create a text file within it.

“file-3.txt”is used here.

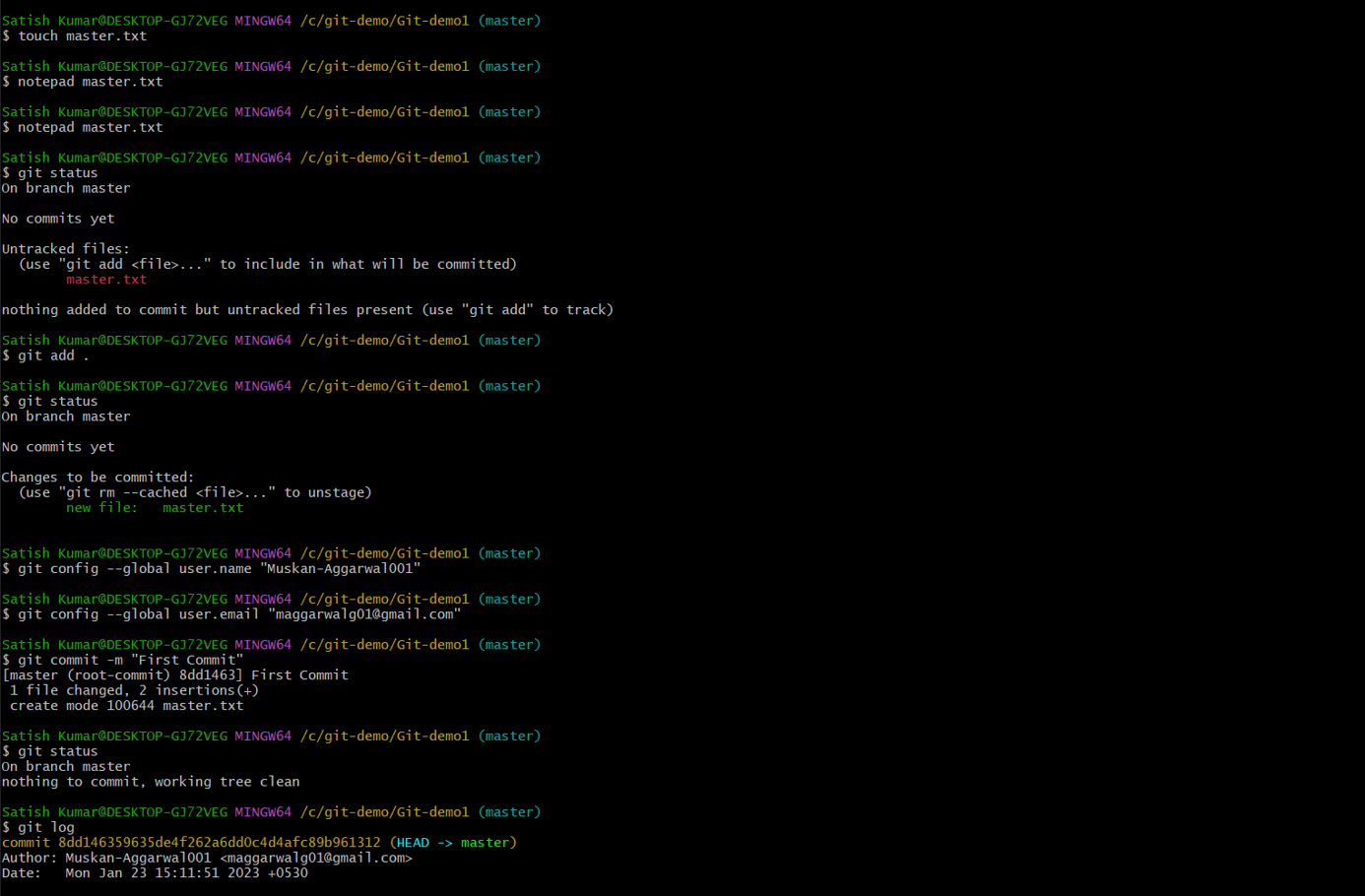
4)**cat>name.txt:** it is used to write the text within the file tha we have created using the touch command.”Ctrl D”command is used to comt out.

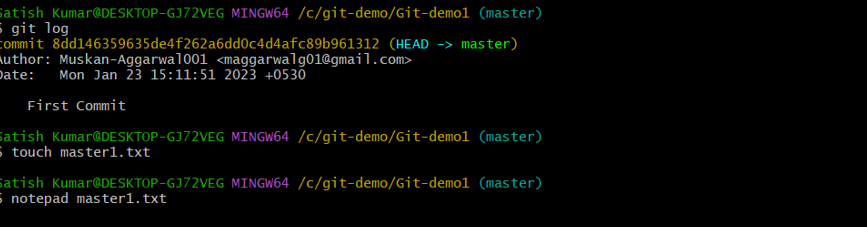
5)**git add name.txt :** it is used to add the changes that have been made in the present working directory .

6)**git commit -m :** it is used to take all the changes that have been made locally and push them upto a remote repository .

7)**git status :** it is used to display state of working directory and staging area .

8)**git log :** it is a utility tool to review and read a history of everything that happen to a repository.



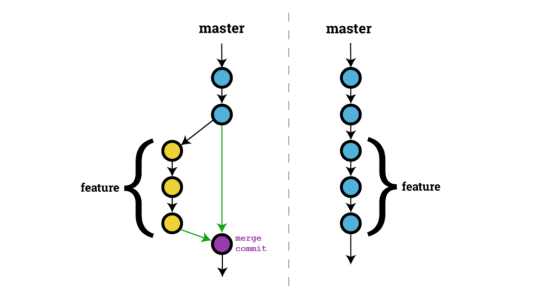


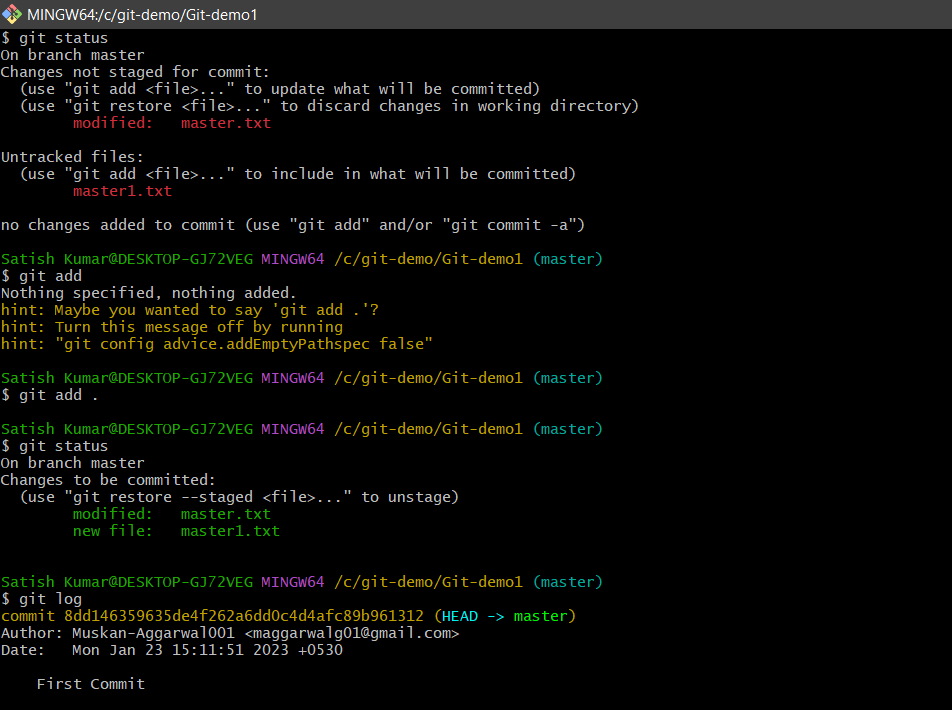
**Experiment 4 Aim: Create and visualize branches**

**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.

1)**git branch :** Git branches are effectively a pointer to a snapshot of your changes .

2)**git checkout :** git checkout command let you to navigate between the branches that we have created by git branch .





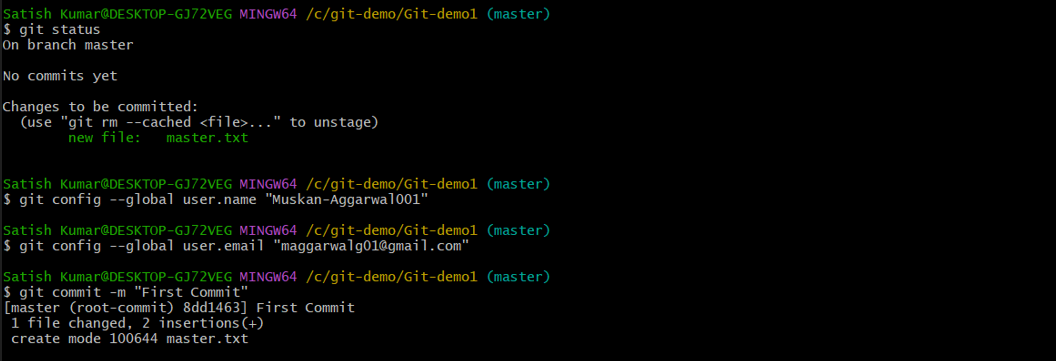
1)**git checkout activity :** it is used to change the branch to the activity .

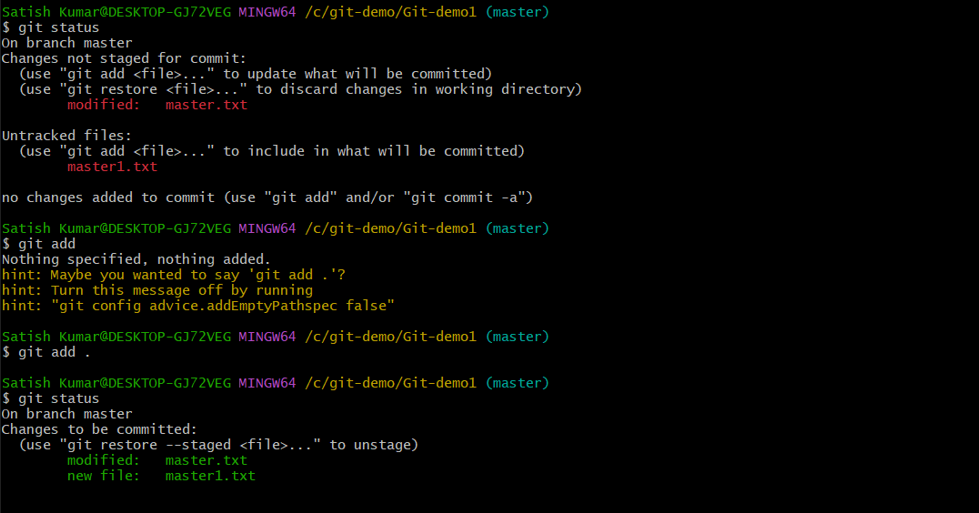
2)**touch file.txt :** it is used to create a file only if the file does not already exist.

3)**cat command :** it is used to add some additional fact to the pre existing file .

4)**git log :** it is used to view repository and all the changes are also visible that we have made.it also tell us the name of author and time.

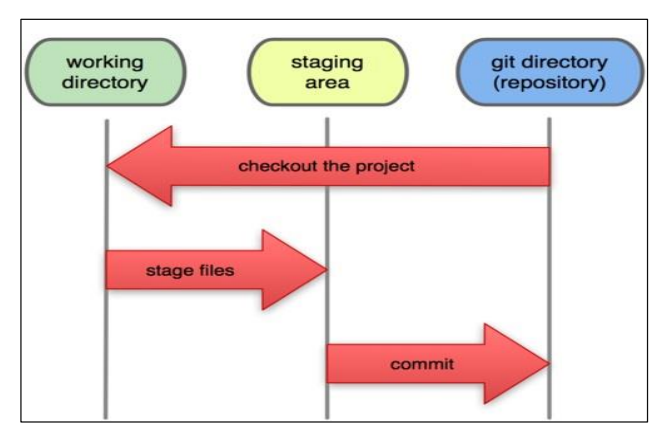
5)**git status :** now it tell us about the file and branch that we are presently working with .





**Experiment 5 Aim: Git lifecycle description**

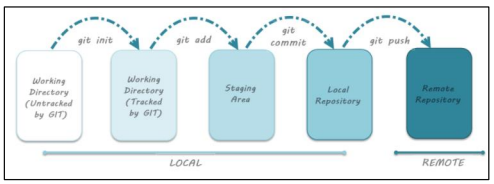
**Theory:** Stages in GIT Life Cycle: Files in a Git project have various stages like Creation, Modification, Refactoring, and Deletion and so on. Irrespective of whether this project is tracked by Git or not, these phases are still prevalent. However, when a project is under Git version control system, 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



1)**git add filename:** This is used to the add a changes in the working directory.

2)**git commit -m :** this is used to print the message . the message must be short and descriptive .

3)**git checkout master :** it is used to change the branch to the master .

4)**git log :** now this we tell us the about all the work that has to be done .

5)**git status :** it tell us that the we are presently working on branch master .

