

Subject Name: **Source Code Management**

Subject Code: **CS181**

Cluster: **Beta**

Department: **CSE**

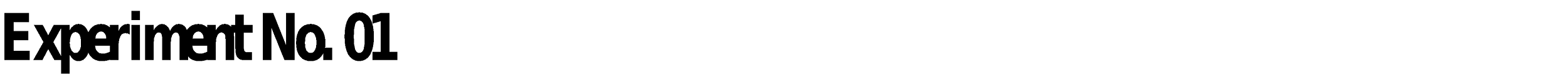


| **Submitted By:**  Aadithya  2110990002  G01-B |  | **Submitted To:**  Dr. Monit Kapoor |
| --- | --- | --- |

**List of Programs**

| **S. No** | **Program Title** | **Page No.** |
| --- | --- | --- |
| 1 | Setting up of git client | 3-5 |
| 2 | Setting up GitHub account | 6-8 |
| 3 | Program to generate logs | 9-12 |
| 4 | Create and visualise branches | 13-15 |
| 5 | Git lifestyle description | 16 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

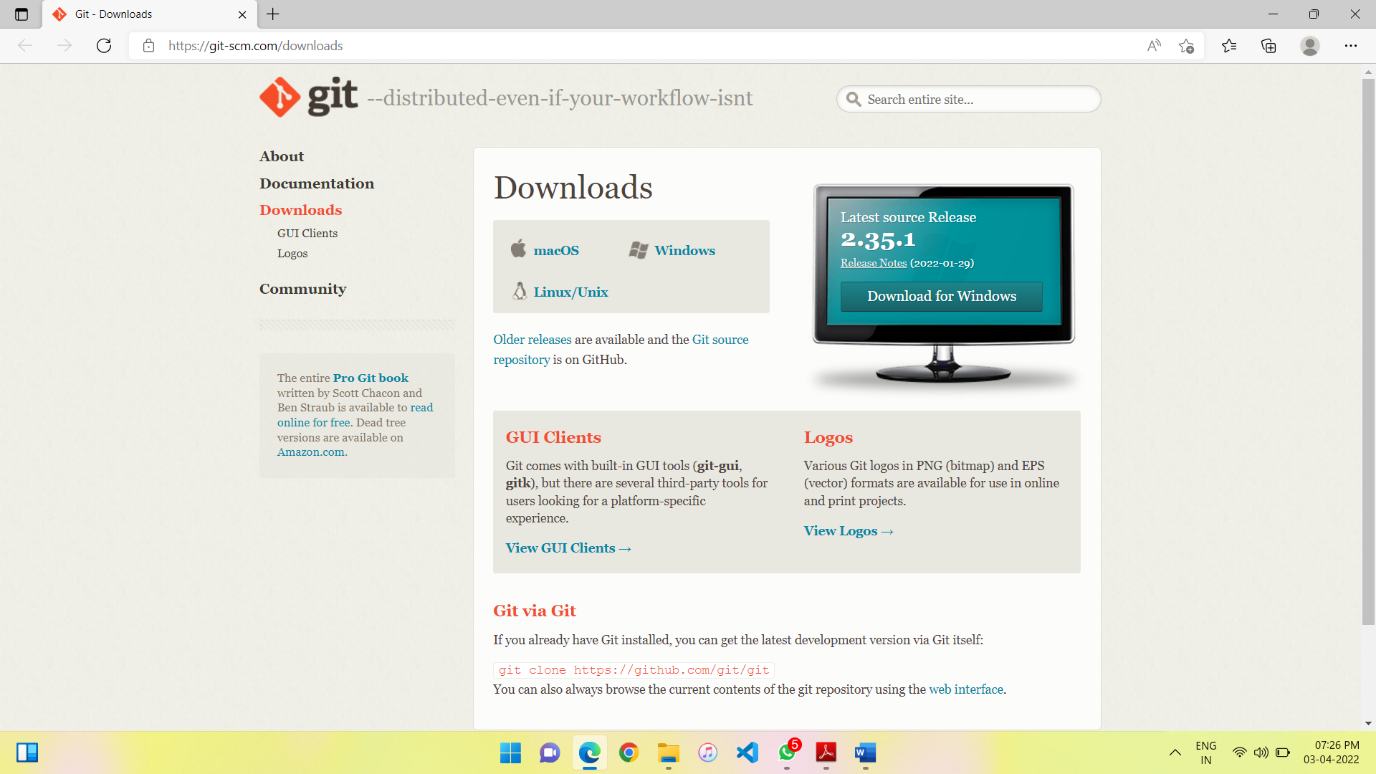
1



**Aim: - Setting up of Git client**

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

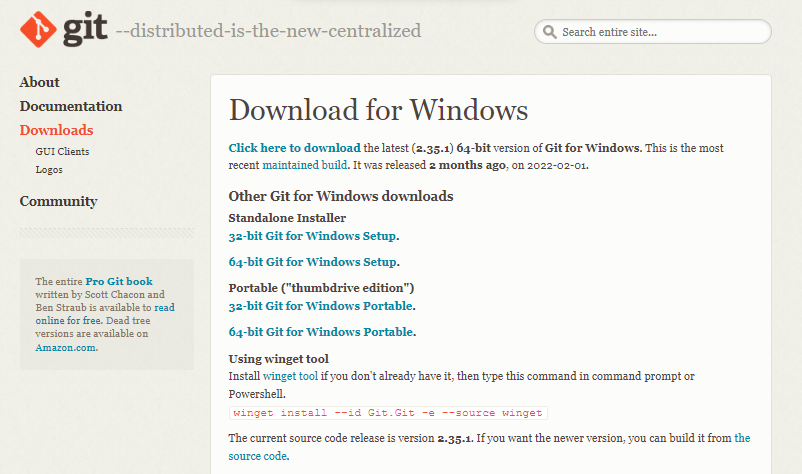
[**https://git-scm.com/downloads**](https://git-scm.com/downloads)



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.

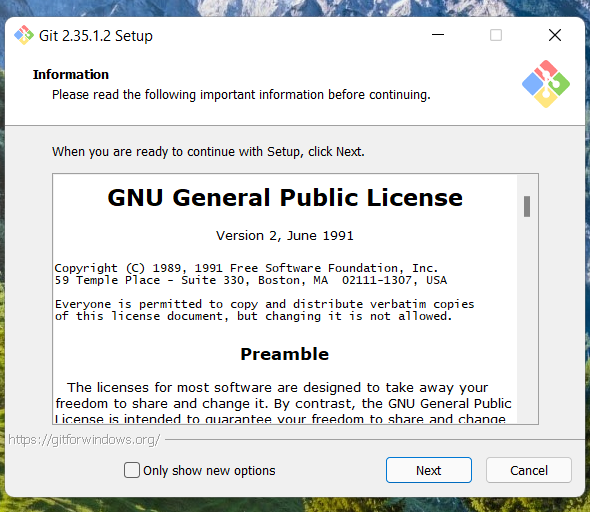


3. Now select the processor of the system you have. (Most of the system are now of 64-bits) After selecting the processor your download will start.

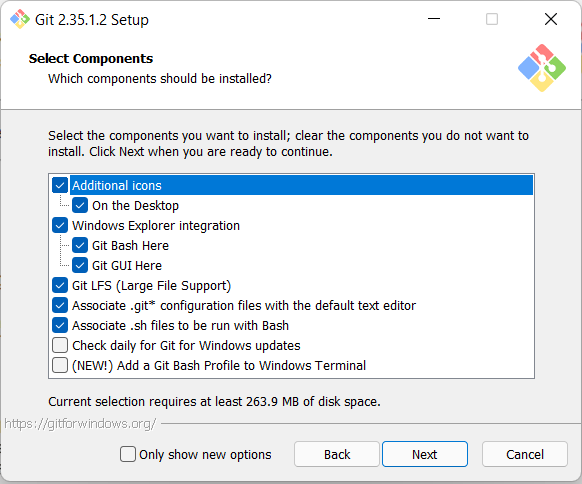
4. Now you have to open this folder.

5. After opening you will give a notification “Do you want to allow this app to make changes in your PC”

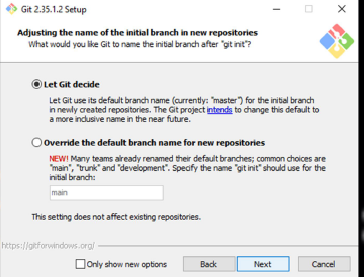
6. Click YES

****

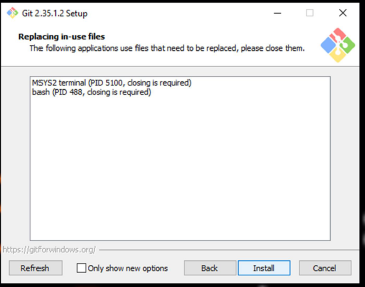
7. Click on Next



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

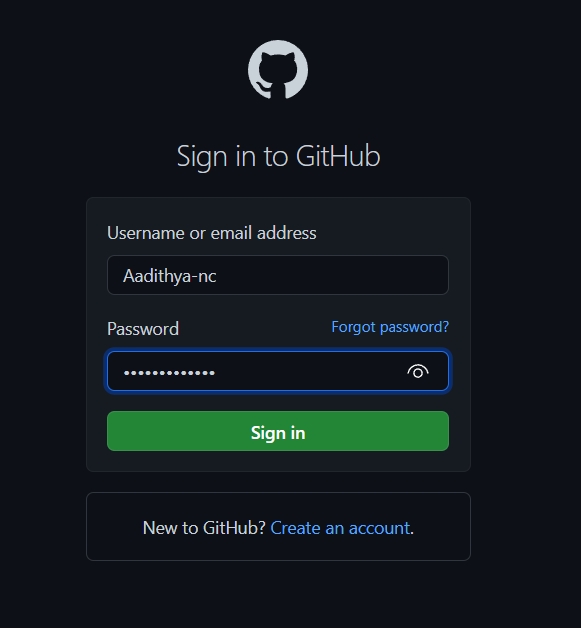
and GitHub account.

EXPERIMENT NO:2 .

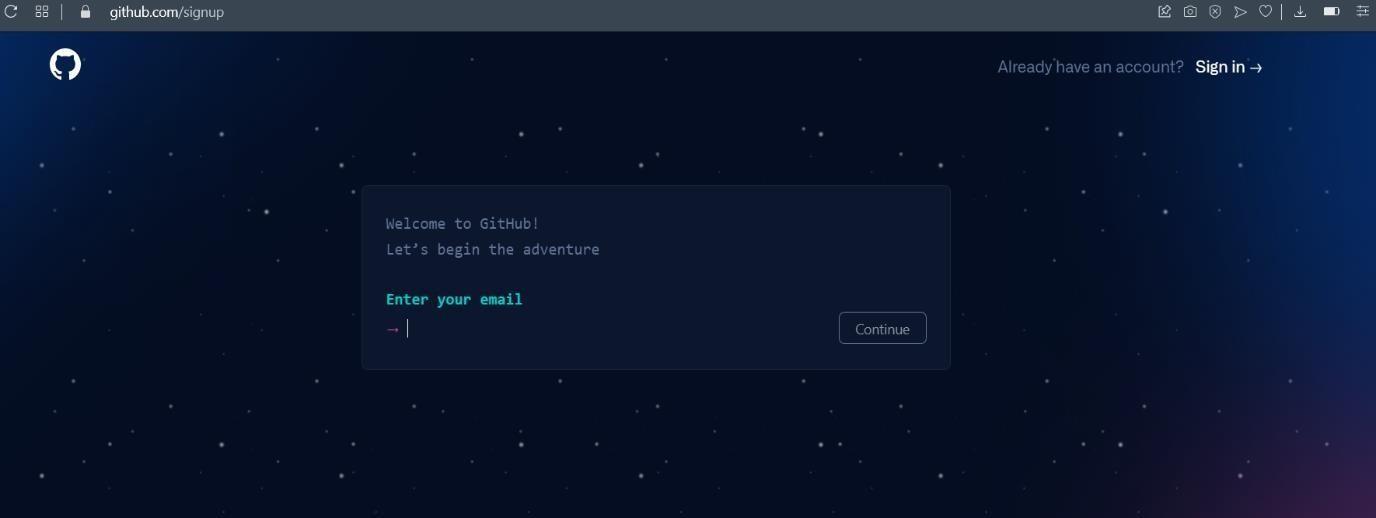
**Aim: - Setting up GitHub Account**

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.



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.



4. Now Click on Create Account.

5. Verify it from your email and you are all set to go.

*EXPERIMENT NO:3 .*

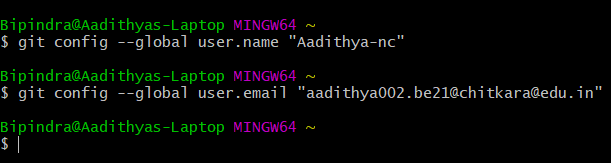
**Aim: - Program to Generate logs**

• 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”**

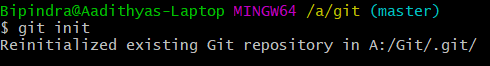
**“Git config --global user. Email “email**”



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 unit” and it creates a folder. git.

**

**Some Important Commands:**

• **ls →** It gives the file names in the folder.

**• ls -lard** → 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.

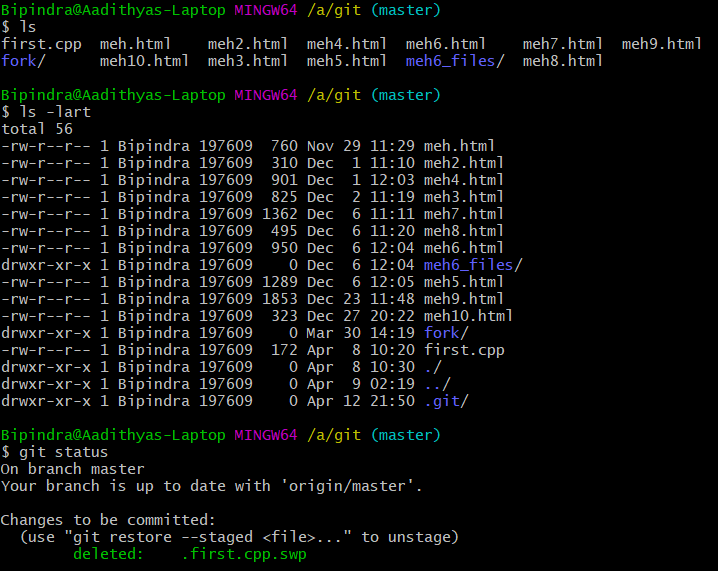
**• rm** -rf. git → It removes the repository.

**• 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

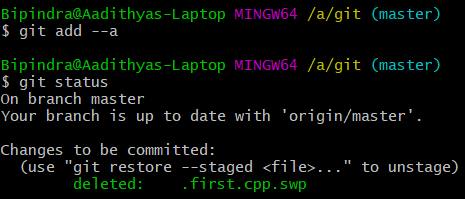
index.html. Now type git status:



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.]

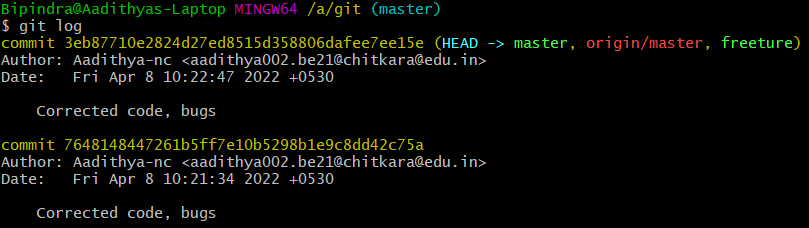


**git commit -m “write any message”** [ For commit the file]

**• 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

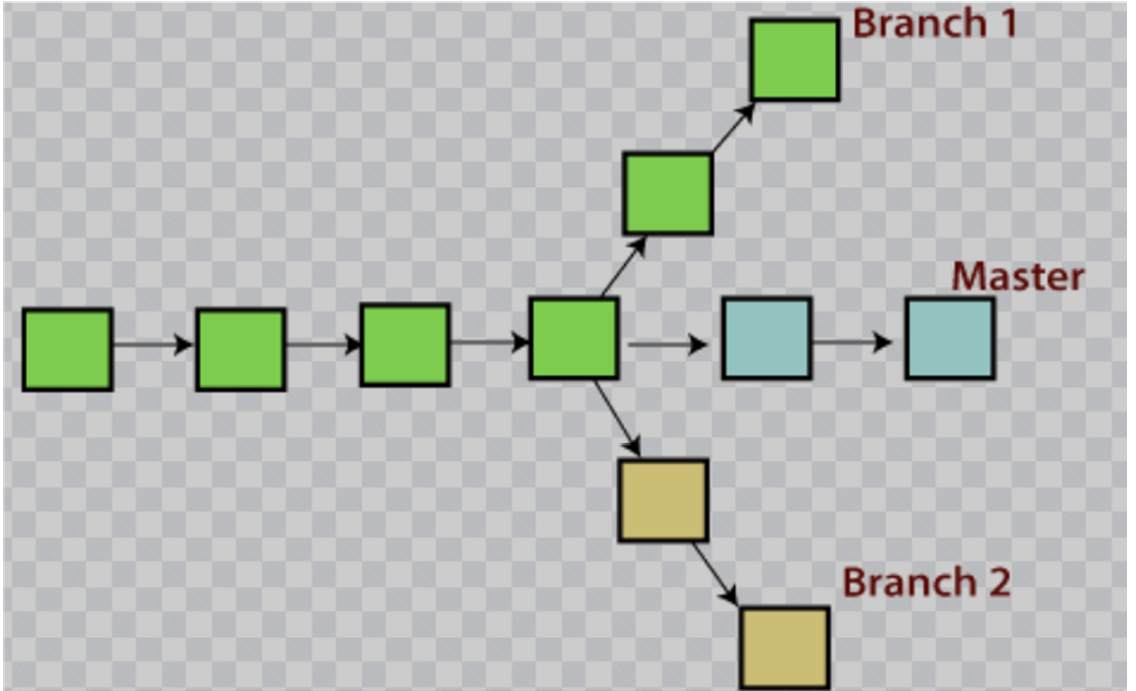
commit message, and other commit metadata.



***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:**

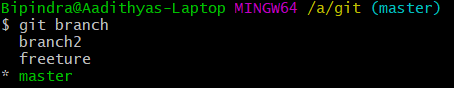
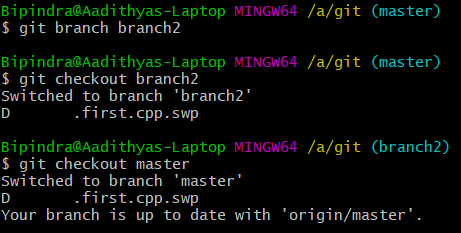
Firstly, add a new branch, let us suppose the branch name is activity1.

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]



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.

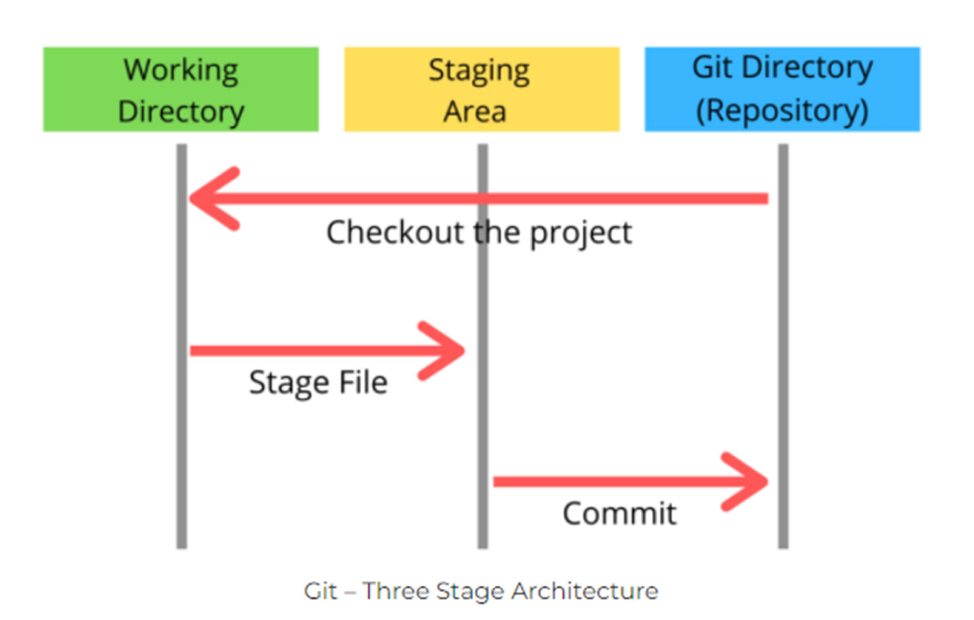
This can be done by using command: -

**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 or project using the

**“Git checkout”** command from this directory.

