Subject Name: **Source Code Management**

Subject Code: **CS181**

Cluster: **Alpha**

Department: **DCSE**

****

|  |  |  |
| --- | --- | --- |
| **Submitted By:**  Shreya  2110991325  G18 |  | **Submitted To:**  Dr. Shikha |



**Experiment No. 01**

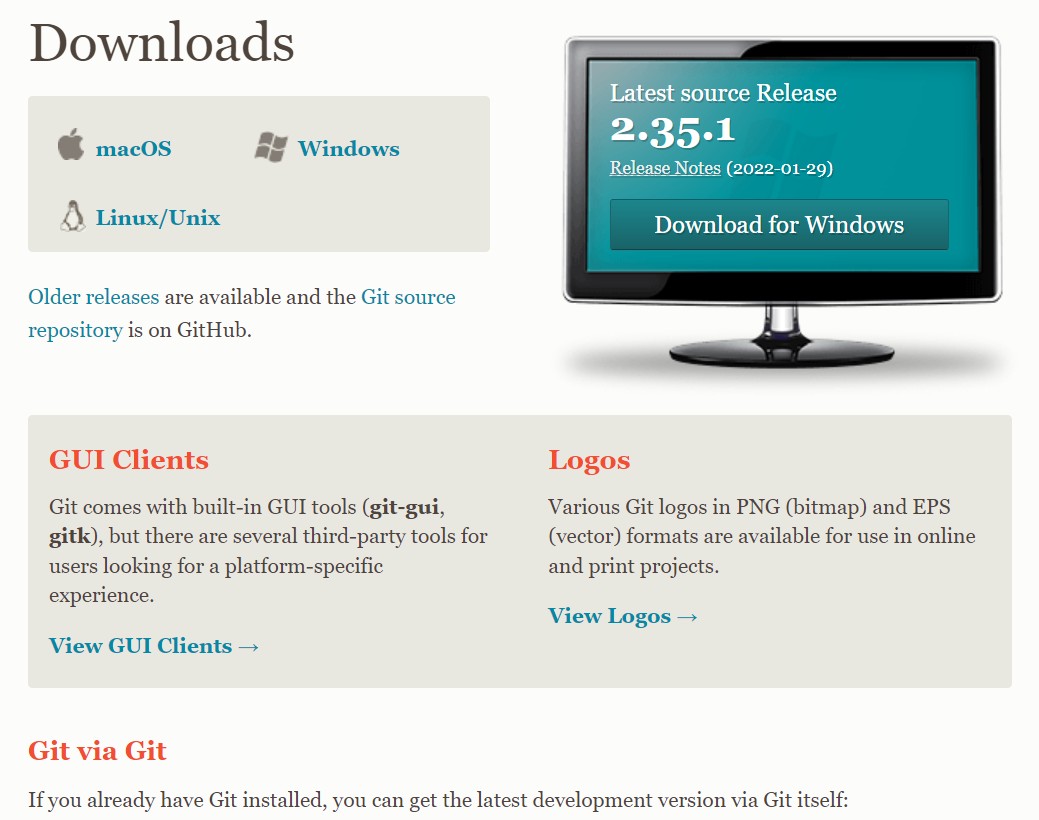
**Aim:** Setting up of Git Client

**Procedure:** The following sections list the steps required to properly install and configure the Git clients i.e. Git Bash and Git GUI on a Windows computer. Git is also available for Linux and Mac. The remaining instructions here, however, are specific to the Windows installation.

**Git installation:**

Download the Git installation program (Windows, Mac, or Linux) from

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



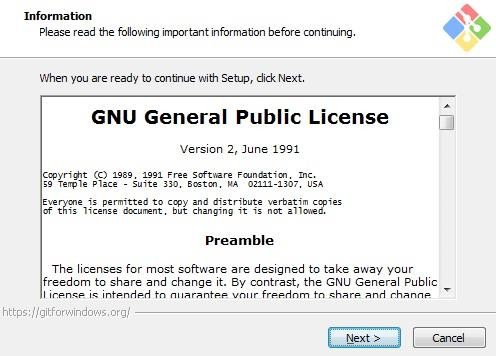
Click on the package given on the page as download 2.23.0 for windows. The download will start after selecting the package.

When running the installer, various screens appear (Windows screens shown). Generally, we can accept the default selections, except in the screens below where we do NOT want the default selections:

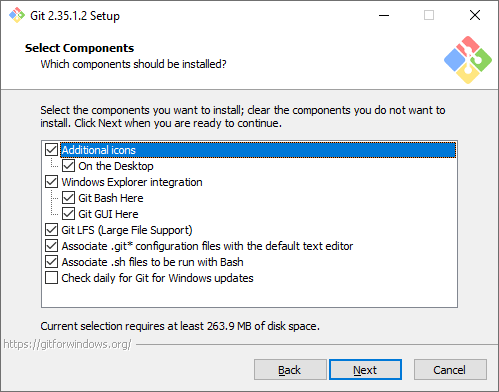
1.Gitinstallation:

**STEP-1**: Download git client from the url(git-scm.com/download)

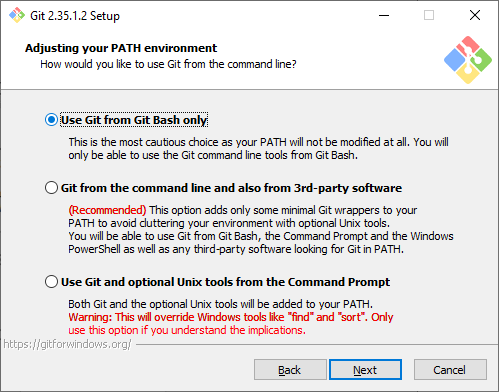
**STEP-2**: Configure and set up git client.

Step 2.1:Click on the download installer file and then click on next. 

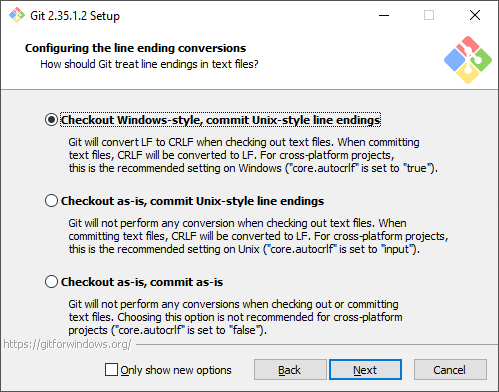
Step 2.2: Simply click on the next button as it automatically selects the required file.



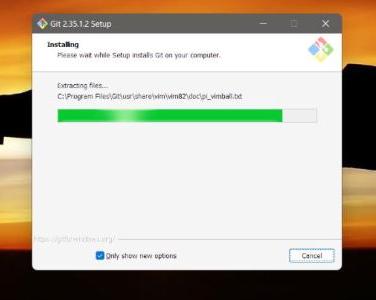
Step 2.3: You can choose your preferred choice. Click next to continue.



Step 2.4: Note: - Just simply click on next as it automatically selects the required file.



Step 2.5: The Git is getting download in your system



**STEP 3**: Open git bash in the file to be edited.

**STEP 4**: Enter t.phe command “git init”in git bash.

//git init initialises the file and creates the master branch by default.

**STEP 5**: enter the command “git status” in git bash.

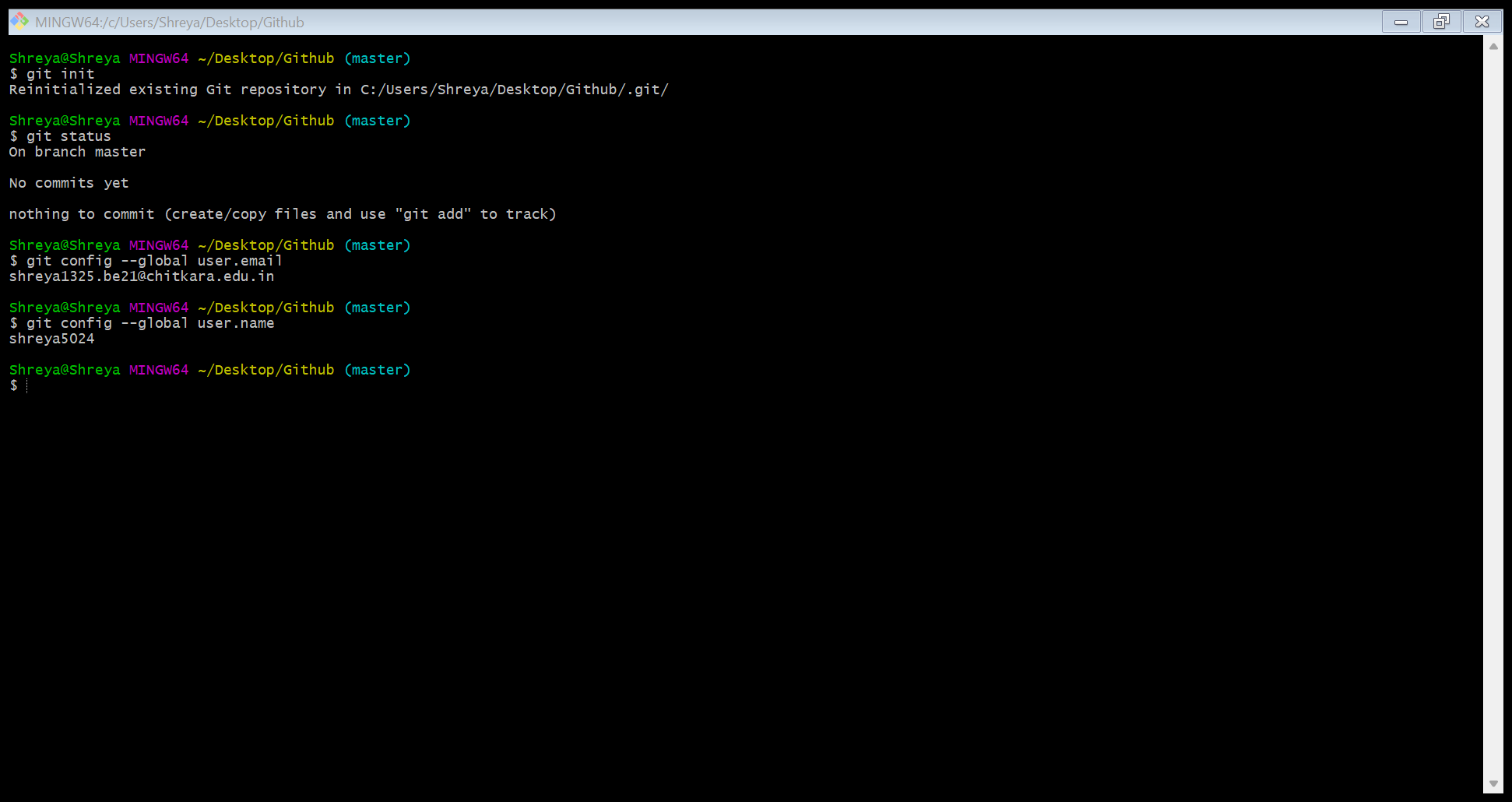
//Git status: The git status command displays the state of the working directory and the staging area.

**STEP 6**: Setting up a user id and name using “git config--global user.name” and

“git config –global user.email”

//enter your email id and username.

Ls: Shows all the files present in the working directory.





**Experiment No. 02**

**Aim:** Setting up GitHub Account

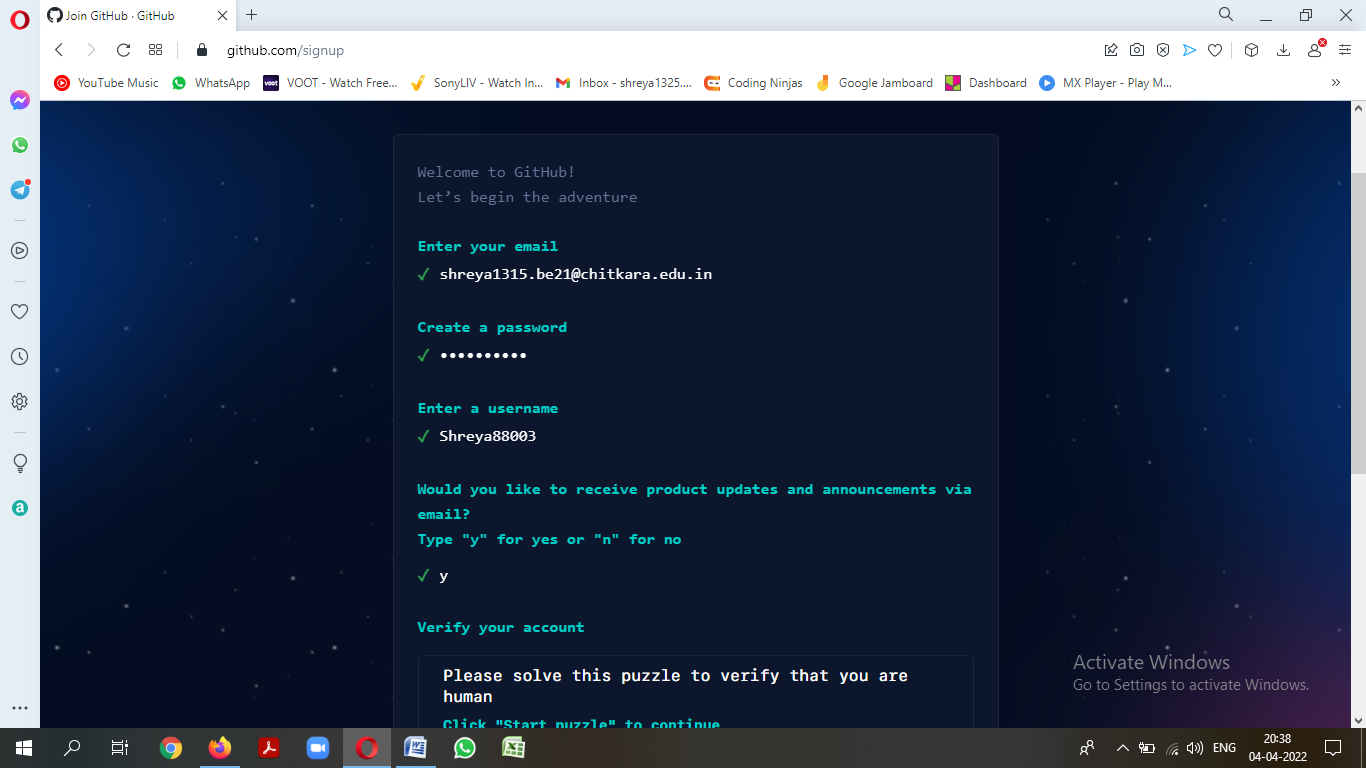
**Procedure:** The following sections list the steps required to properly install and configure the Git clients - Git Bash and Git GUI

- on a Windows 7 computer. Git is also available for Linux and Mac. The remaining:

1. Creating an account

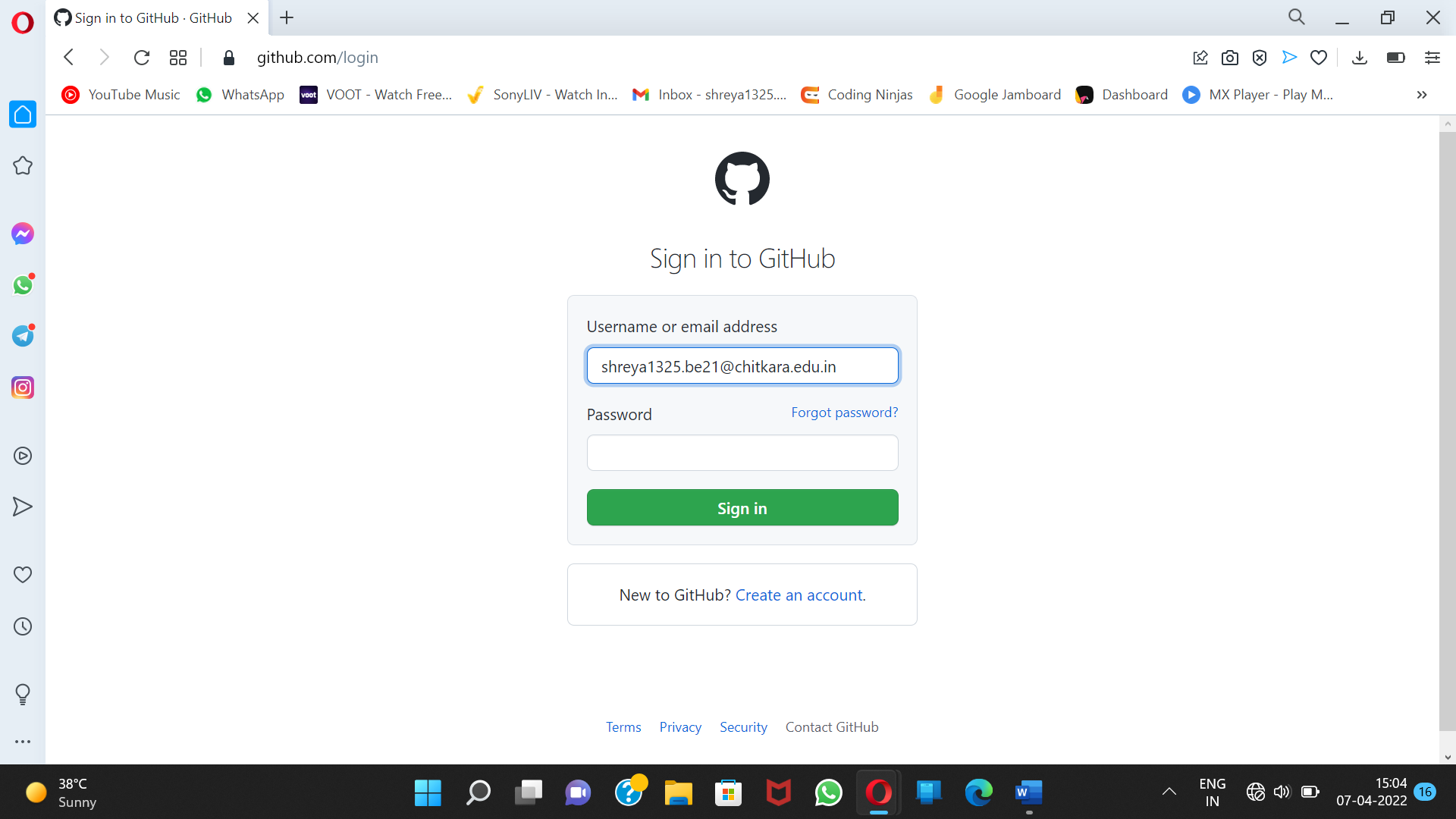
To sign up for an account on GitHub.com, navigate to <https://github.com/> and follow the prompts.

2. To keep our GitHub account secure we should use a strong and unique password. For more information, see "[Creating a strong password](https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/creating-a-strong-password)."



3. Verifying our email address

To ensure we can use all the features in our GitHub plan, verify our email address after signing up for a new account. For more information, see "[Verifying our email address](https://docs.github.com/en/github/getting-started-with-github/signing-up-for-github/verifying-your-email-address)."





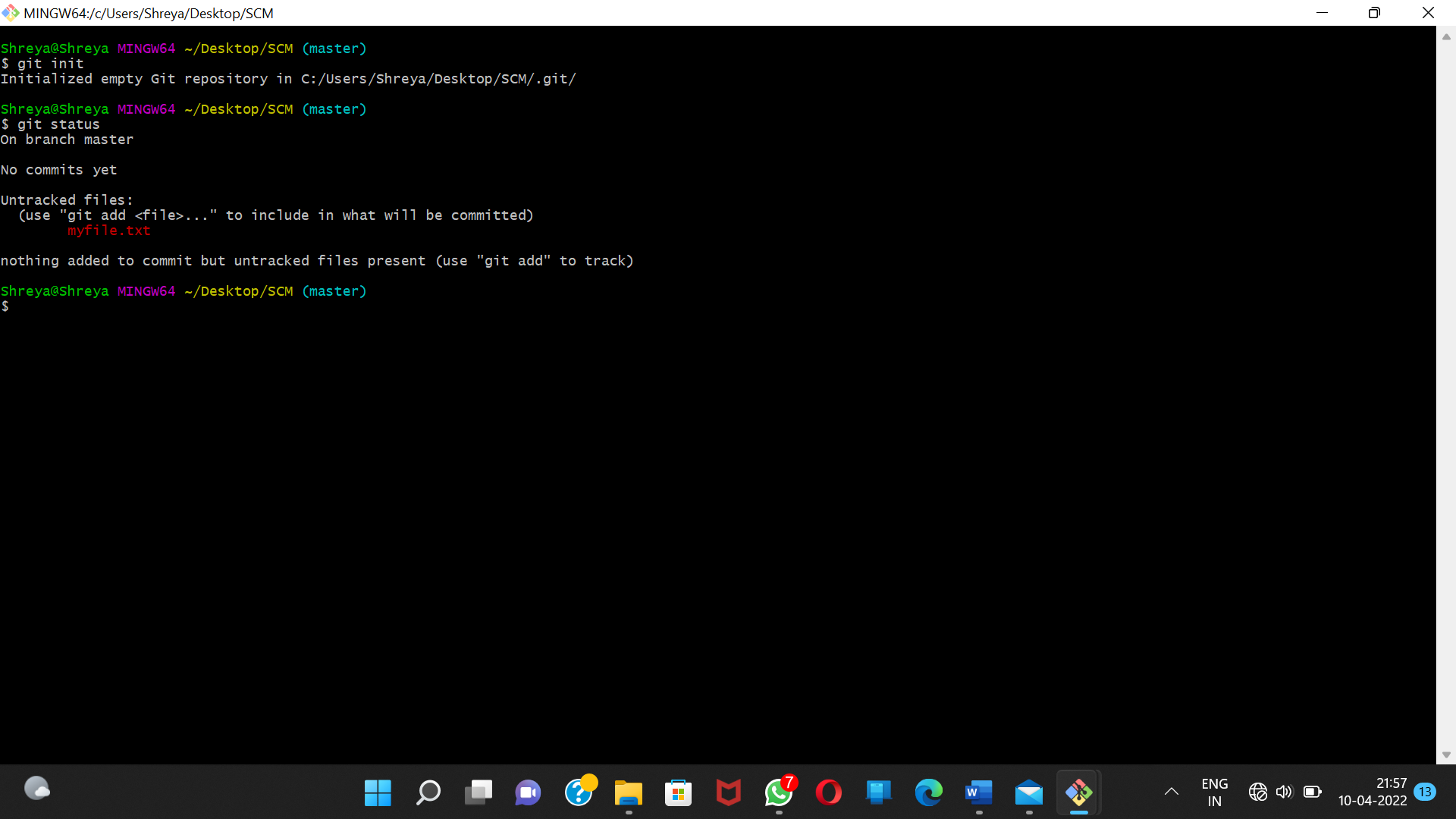
**Experiment No. 03**

**Aim:** Program to generate logs

**Procedure:** In log history we can see the changes done by whom and when i.e. shows exact time, date and author name by whom changes has been done and what change has been done as shown in below screenshot.

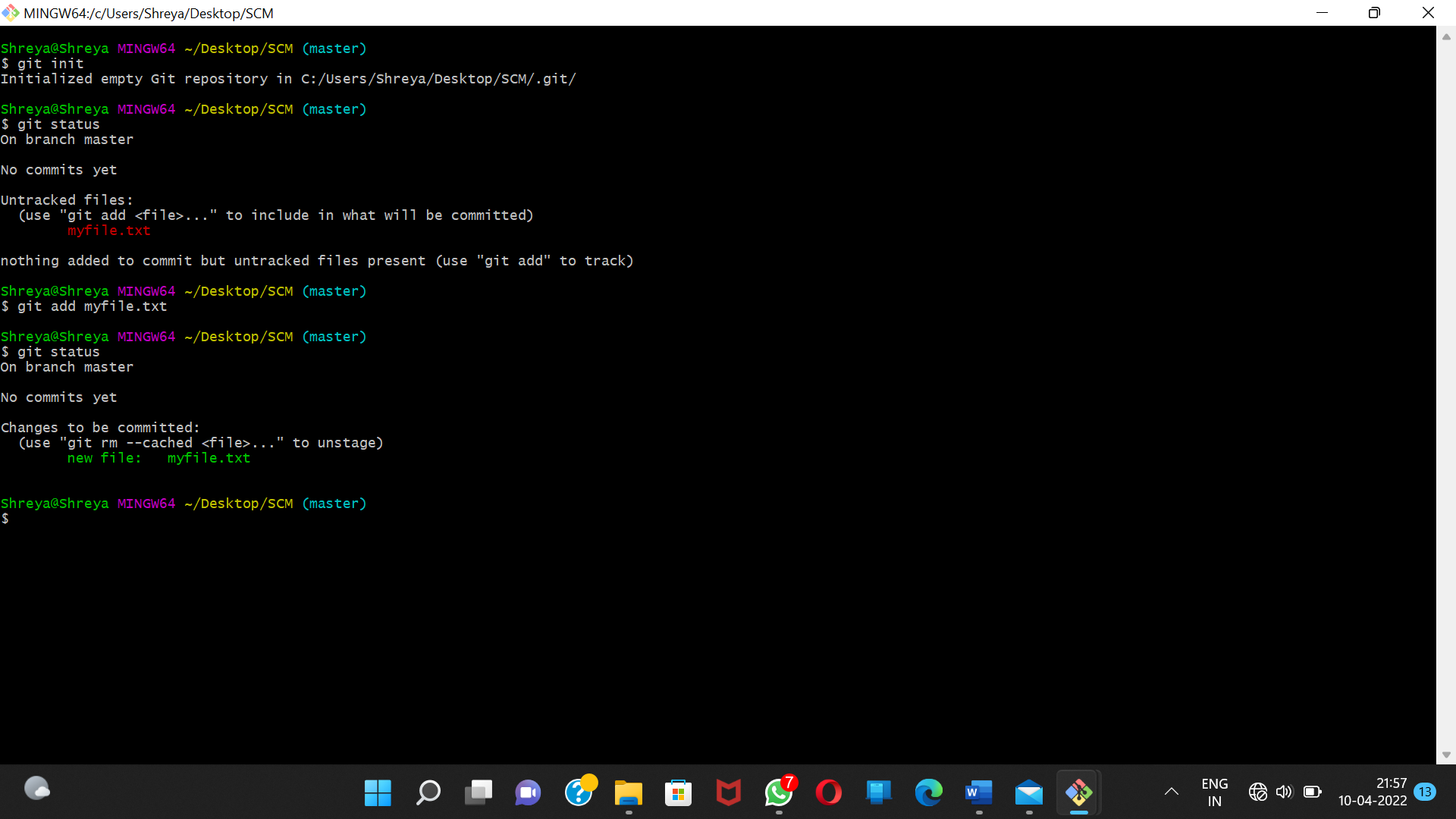
**Step 1**: Create a file myfile.txt.

//the file is showing untracked on entering the command “git status” because we have not added the file in staging area.



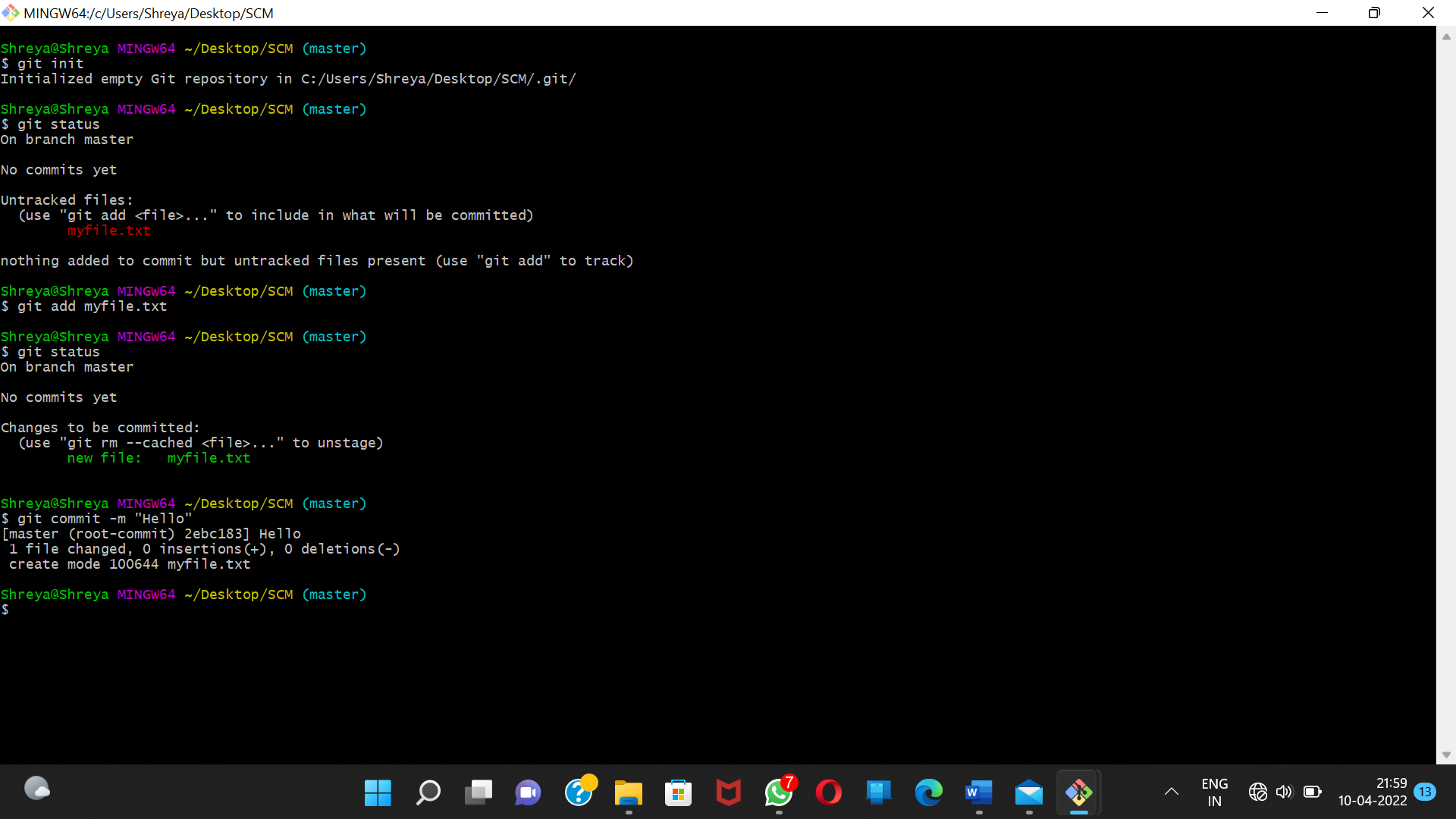
**Step 2**: Enter the command “git add ”

//this command will make this file trackable by git.Thus when we do git status, the status of file comes tracked



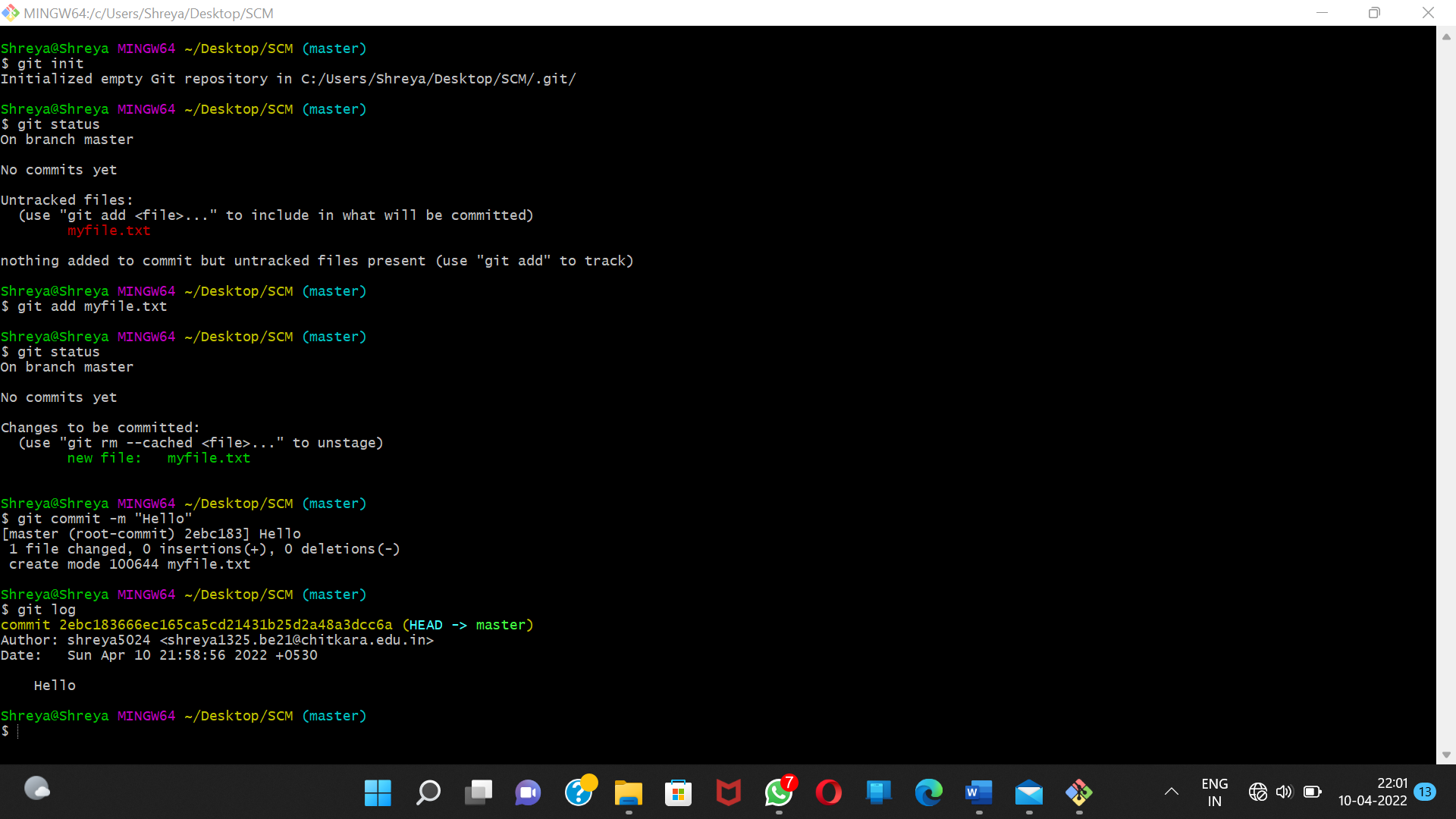
**Step 3**: Enter the command “git commit -m “.hello”.

//this command will commit the changes i.e. the file aikshit.txt will be committed.



**Step 4**: Enter the command “git log”

//this command will show all the changes made till this moment.





**Experiment No. 04**

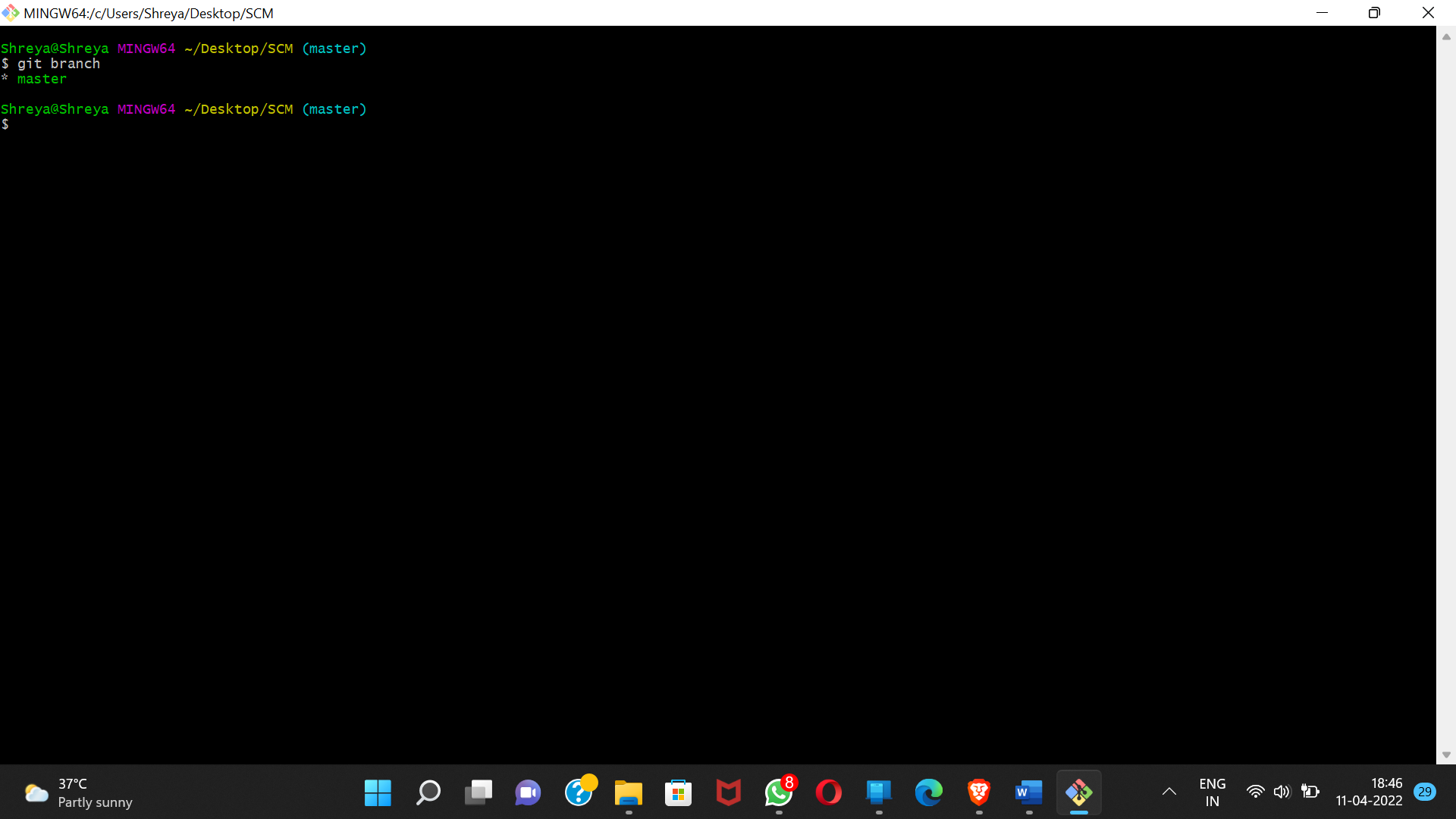
Aim: Create and visualize branches

**Procedure:** A branch in Git is simply a lightweight movable pointer to one of these commits. The default branch name in Git is master.

Step 1: Checking branches

You can check which branch you are working in by using the command

‘git branch’

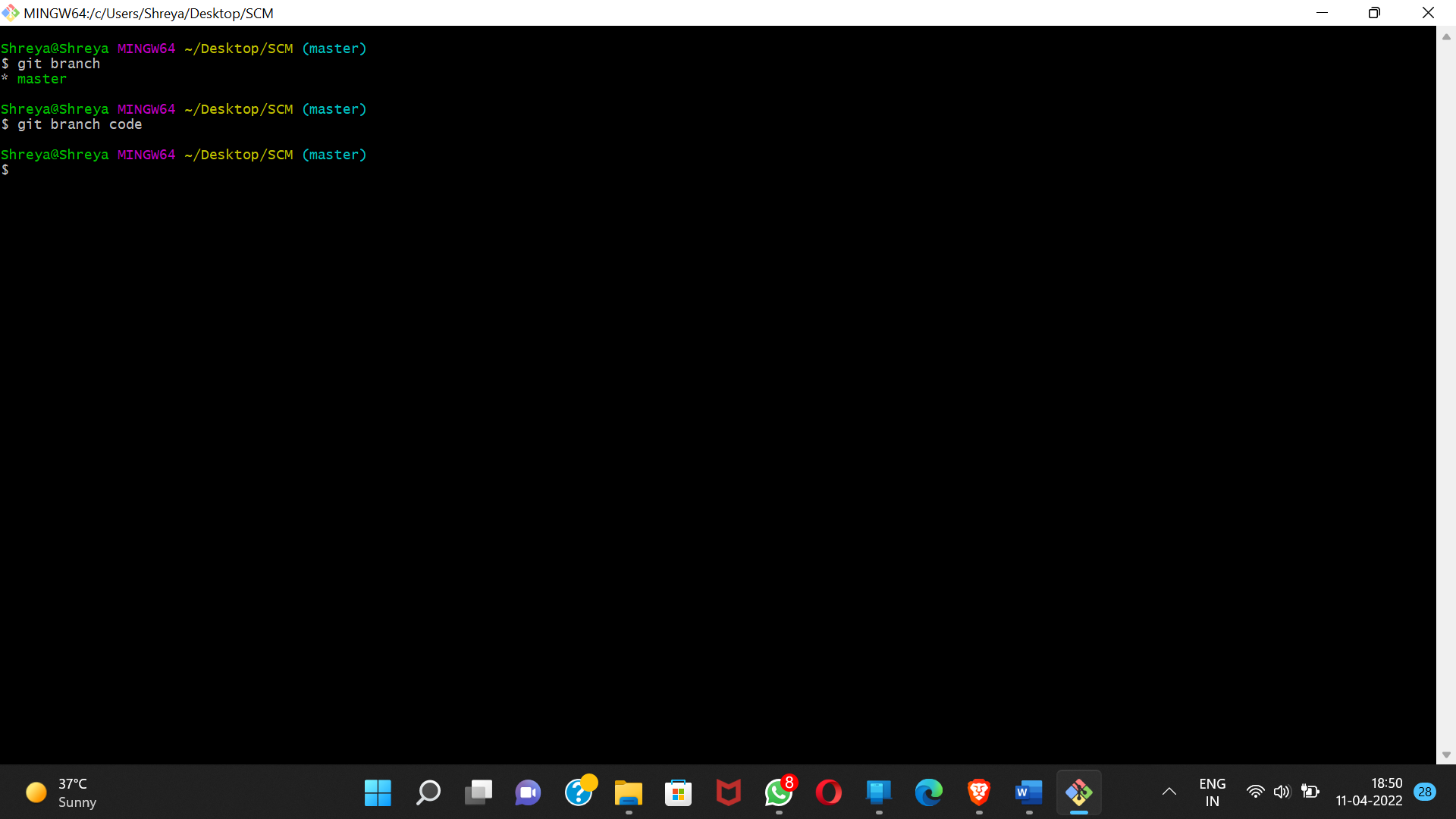


The default branch is always the master branch.

Step 2: Creating multiple branches

You can create multiple branches. You can do it by typing:-

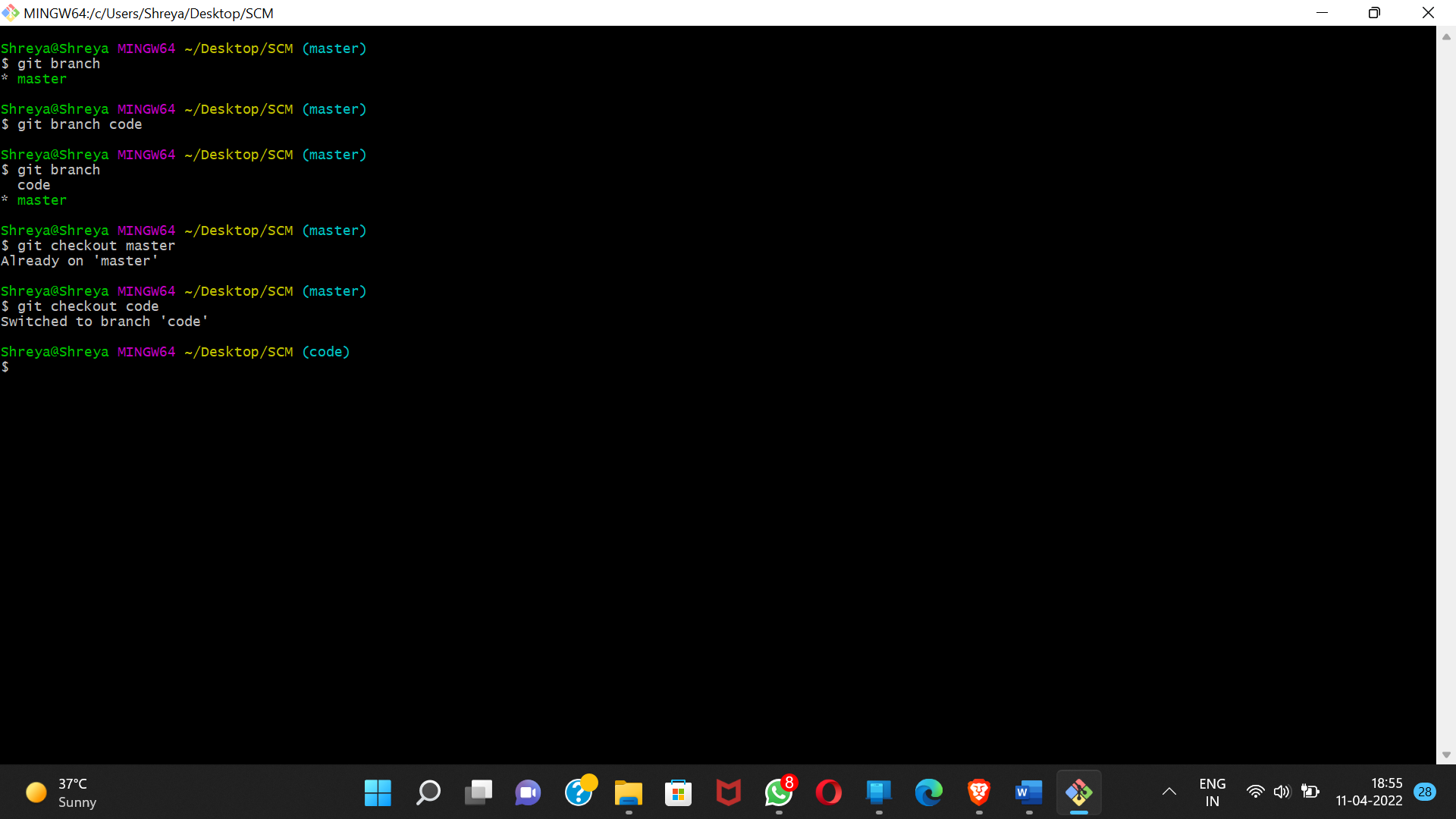
git branch (BRANCH NAME)

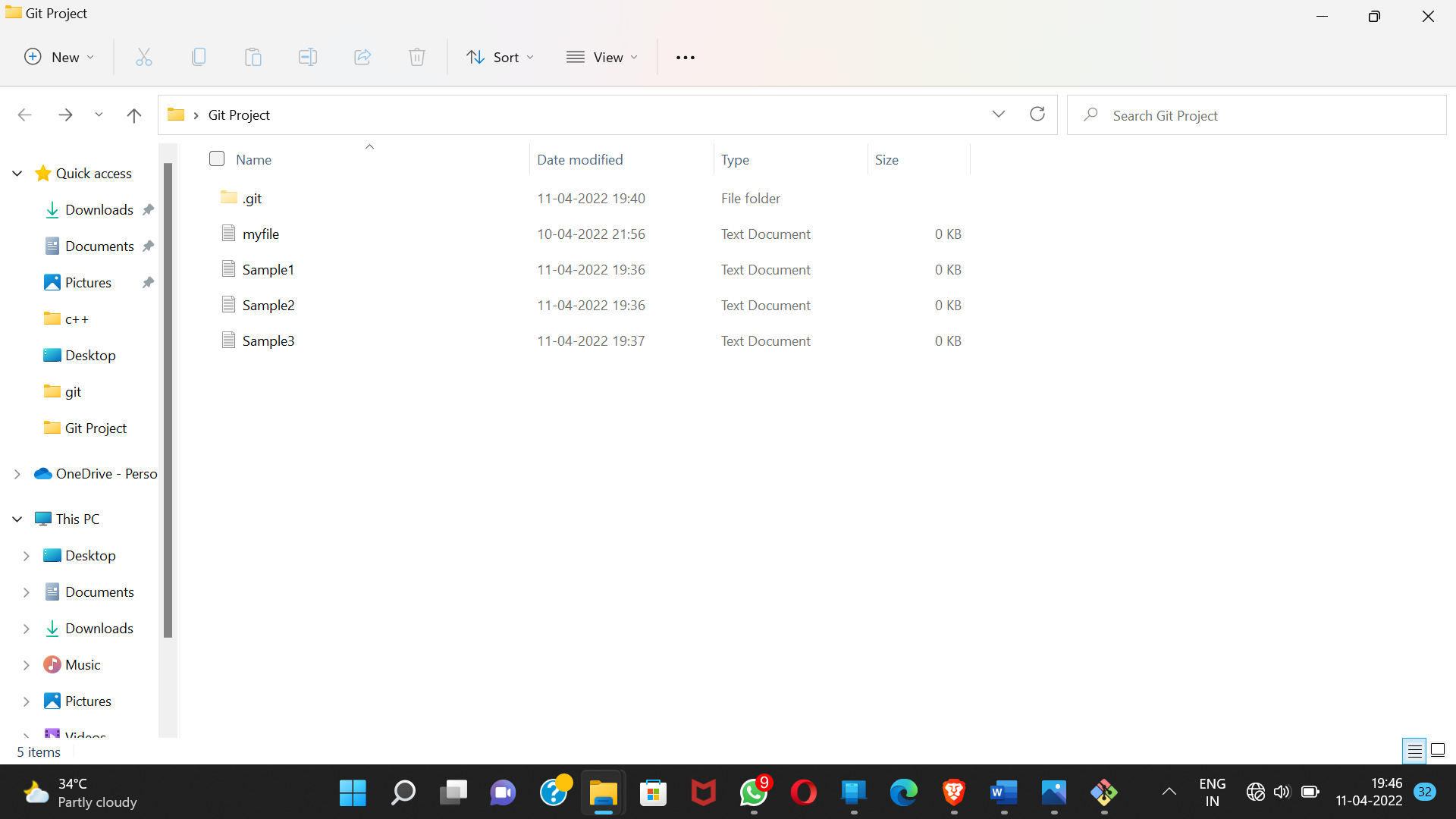


Step 3: Changing branches

To switch to the other branch. You can do it by typing:-

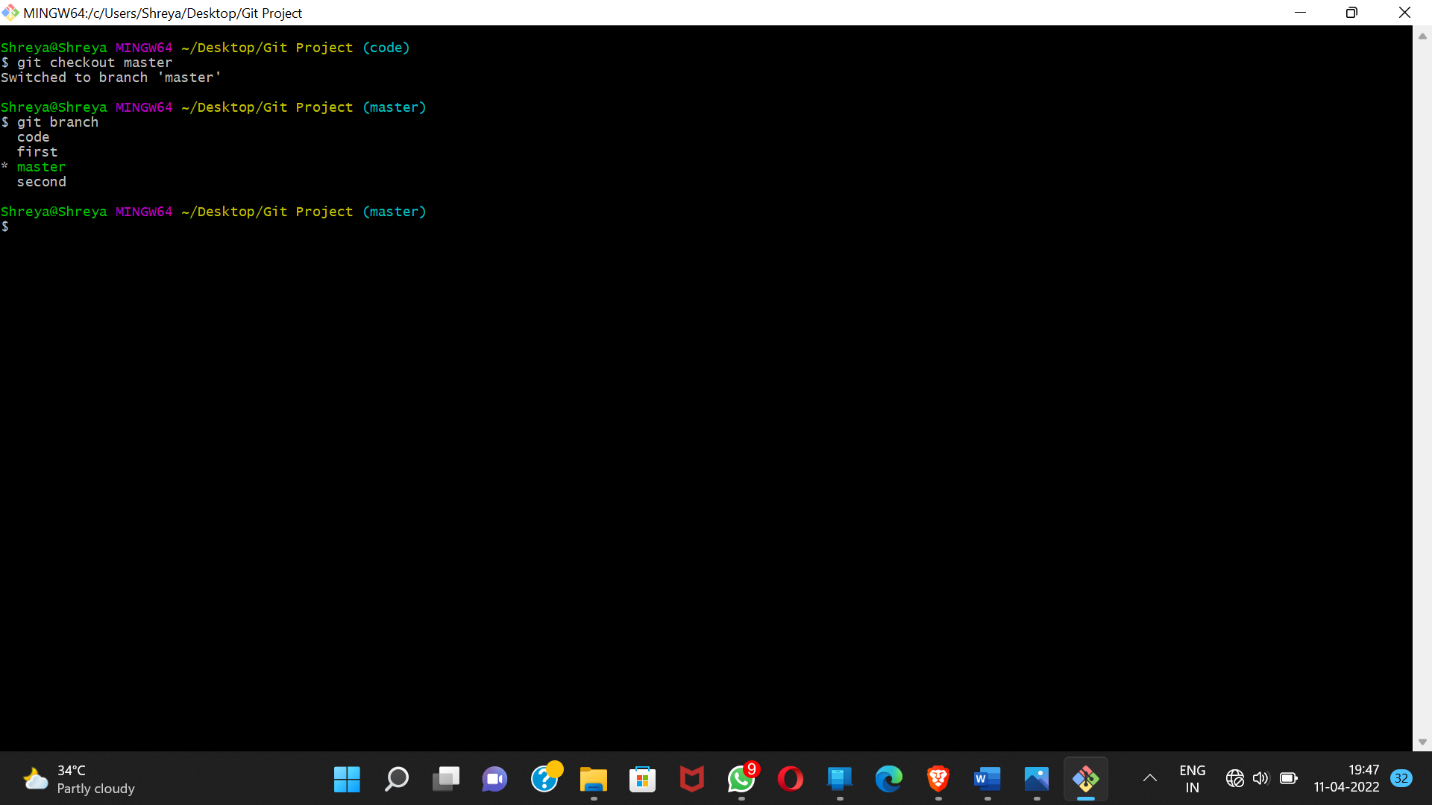
git checkout (BRANCH NAME)



Step 4: Now add file to the new branch and commit it. 

Step 5: Now switch to branch and check file.

Now you can see that there is no file named HelloWorld.txt in the master branch because we created the file in the sample1 branch. So, it will be exclusive to the feature1 branch.

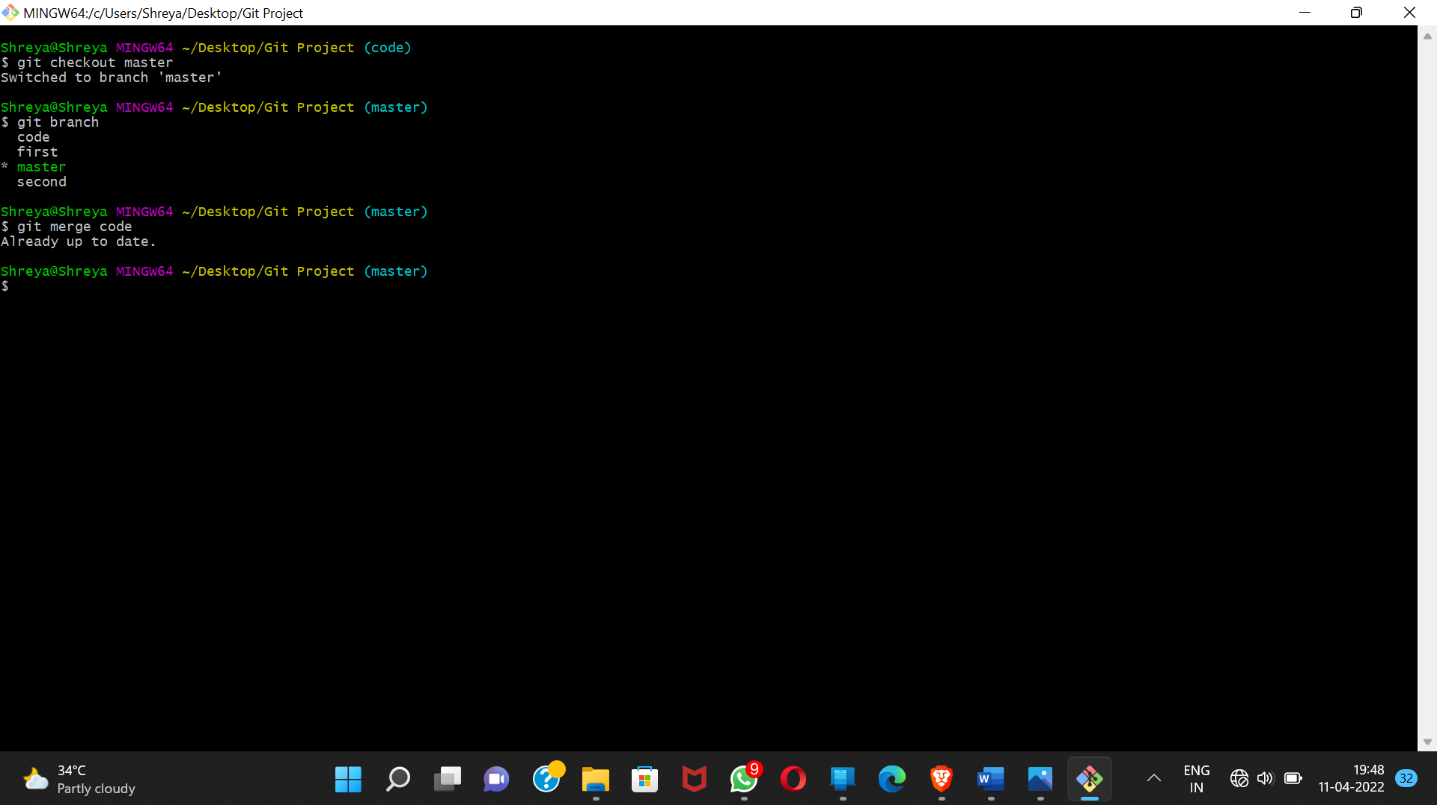


Step 6: Git Merging

Now you can merge two branches by command.

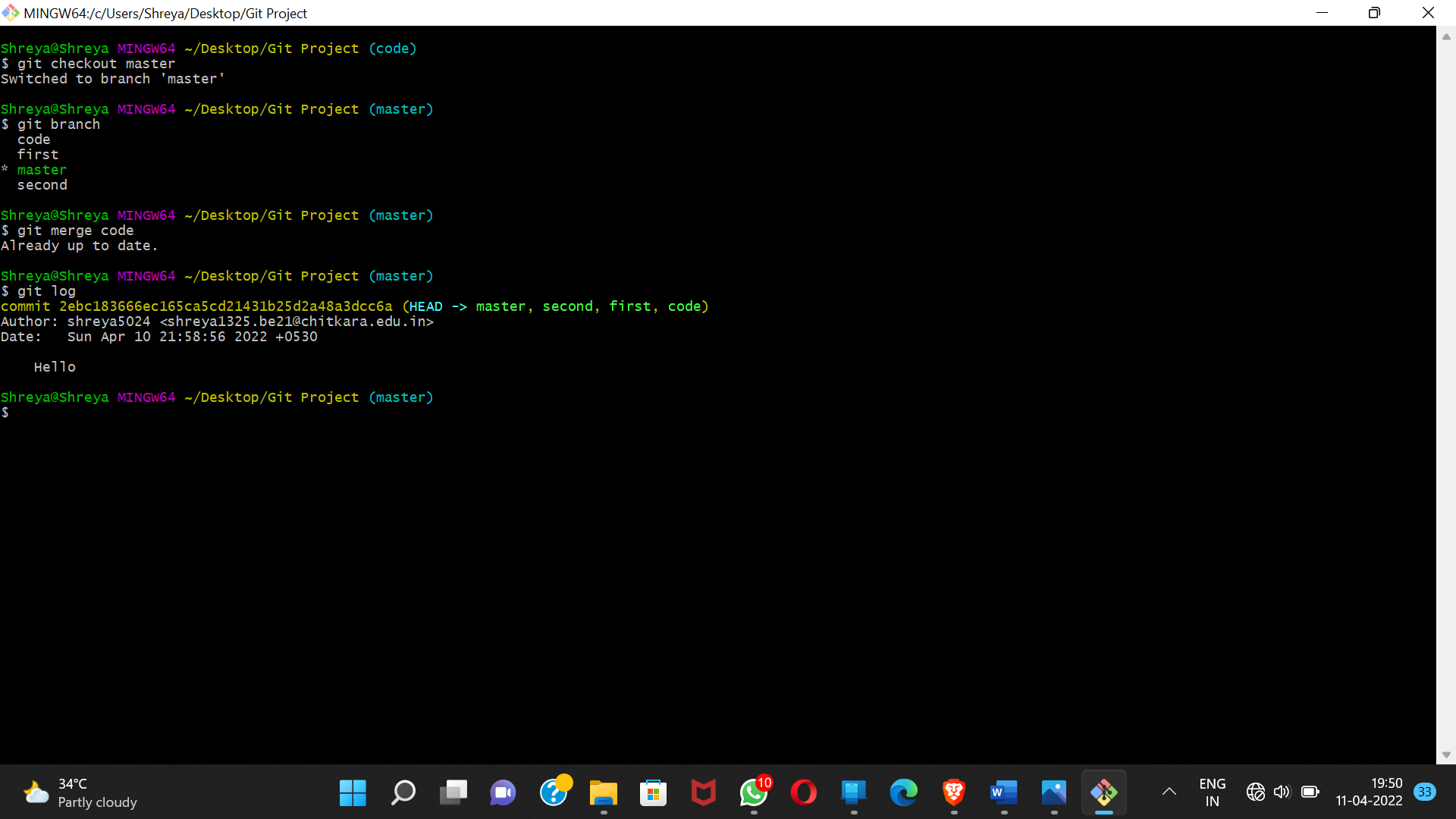
git merge (BRANCH NAME)

If you want to merge a new branch in master branch you need to first checkout into the master branch and then run the command.



Step7: Running Git log

By running git log command on the master branch you can see all the commits made in master as well as the sample1 branch.





**Experiment No. 05**

Aim: Git Lifestyle

Procedure: There are three stages for git lifecycle:

1) Working directory

2) Staging area

3) Git repository

Working Directory:

The working directory is the folder in your local computer where the project files and folders are stored.

The local directory is created by the command ‘git init’ which creates a ‘.git’ named folder which is used to track the files in the directory.

‘.git folder’ is generally hidden but can be tracked enabling hidden

files.

Staging area:

The staging area has those files which are supposed to go to the next commit. Only those files which are needed to go to the next commit stay in the staging area.

You can shift the files to the git repository by using the command

‘git add --a’.

Git repository:

Now since we have all the files that are to be tracked and are ready in the staging area, we are ready to commit our files using the git commit command. Commit helps us in keeping the track of the metadata of the files in our staging area. We specify every commit with a message which tells what the commit is about.

You can commit files by using command ‘git commit -m “message”’

