|  |
| --- |
| Text  Description automatically generated with medium confidence |

**SOURCE CODE MANAGEMENT FILE**

­

***SUBMITTED TO:***

Dr.Chetna Kaushal

(Department of Computer Science and Engineering)

Chitkara University Institute OF Engineering & Technology

RAJPURA , PUNJAB

***SUBMITTED BY:***

NAME: KASHISH GOYAL

ROLL NO. : 2210990496

GROUP : 24(X) Group-B

**SOURCE CODE MANAGEMENT FILE**

SUBJECT NAME: SOURCE CODE MANAGEMENT SYSTEM (SCM)

SUBJECT CODE: 22CS003

SESSION: 2022-23

DEPARTMENT: DCSE

**SUBMITTED BY:**

NAME: KASHISH GOYAL

R NO.: 2210990496

GROUP: G24-X(Group-2)

**\*LIST OF PROGRAMS\***

|  |  |  |
| --- | --- | --- |
| **S. No** | **Program Title** | **Page No.** |
| 1 | Setting up of Git Client | 4-7 |
| 2 | Setting up GitHub Account | 8-9 |
| 3 | Generate logs | 10-13 |
| 4 | Create and visualize branches | 14-15 |
| 5 | Git life cycle description | 16-18 |
| 6 | Add collaborators on GitHub Repo |  |
| 7 | Fork and Commit |  |
| 8 | Merge and Resolve conflicts created due to own activity and collaborators activity. |  |
| 9 | Reset and revert |  |

**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 previous changes in a 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:

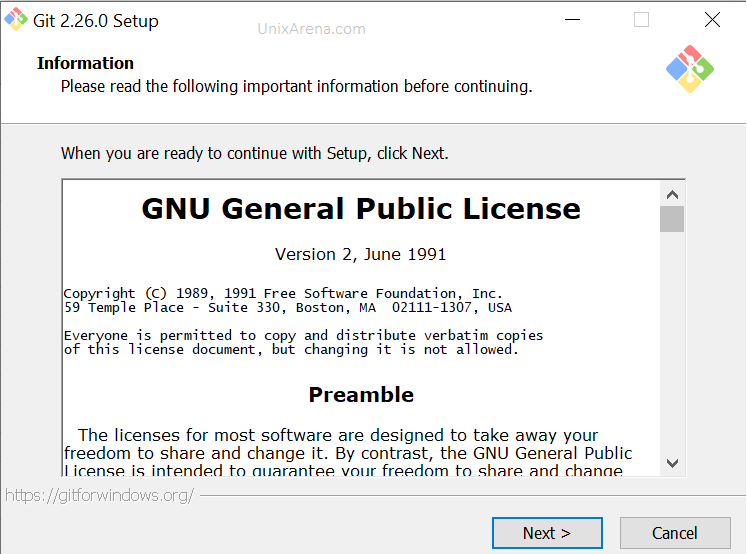
We can install Git on windows, using the most official build which is available for download on the GIT’s official website or just typing (scm git) on any search engine .We can go on this site <https://git-scm.com/download> and can select the platform and bit-version to download . And after clicking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

SNA

PSHOTS OF DOWNLOAD:

Graphical user interface

Description automatically generated with medium confidence



**GIT SETUP**

Graphical user interface, text, application, email

Description automatically generated

**GIT INSTALLATION**

**GIT BASH LAUNCHED**

**Text

Description automatically generated**

**EXPERIMENT 2**

AIM: Setting up GitHub account.

THEORY:

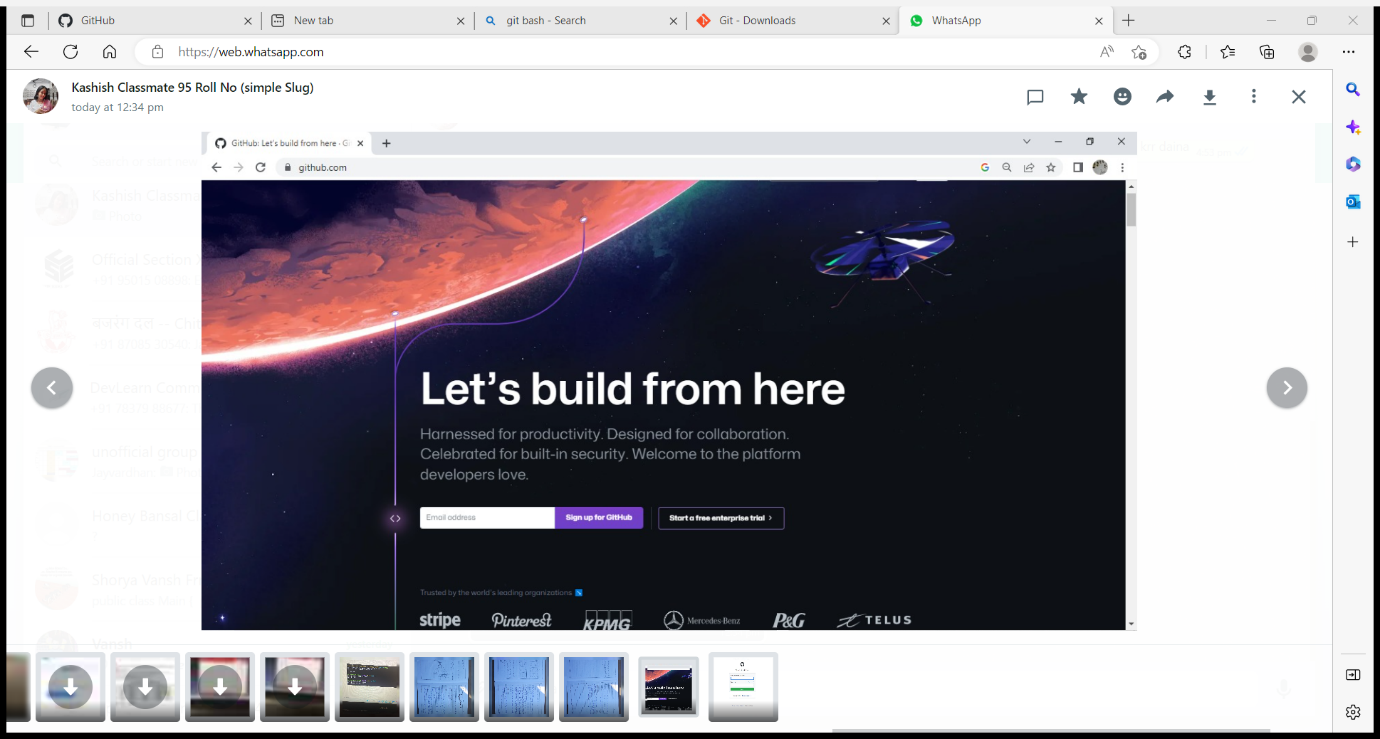
GitHub : GitHub is a website and cloud-based service (client) that helps an individual or developers to store and manage their code .We can also track as well as control changes to our or public code.

Advantages of GitHub: GitHub has a user-friendly interface add 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 with 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

IMAGES





**EXPERIMENT 3**

AIM: program to generate log.

THEORY:

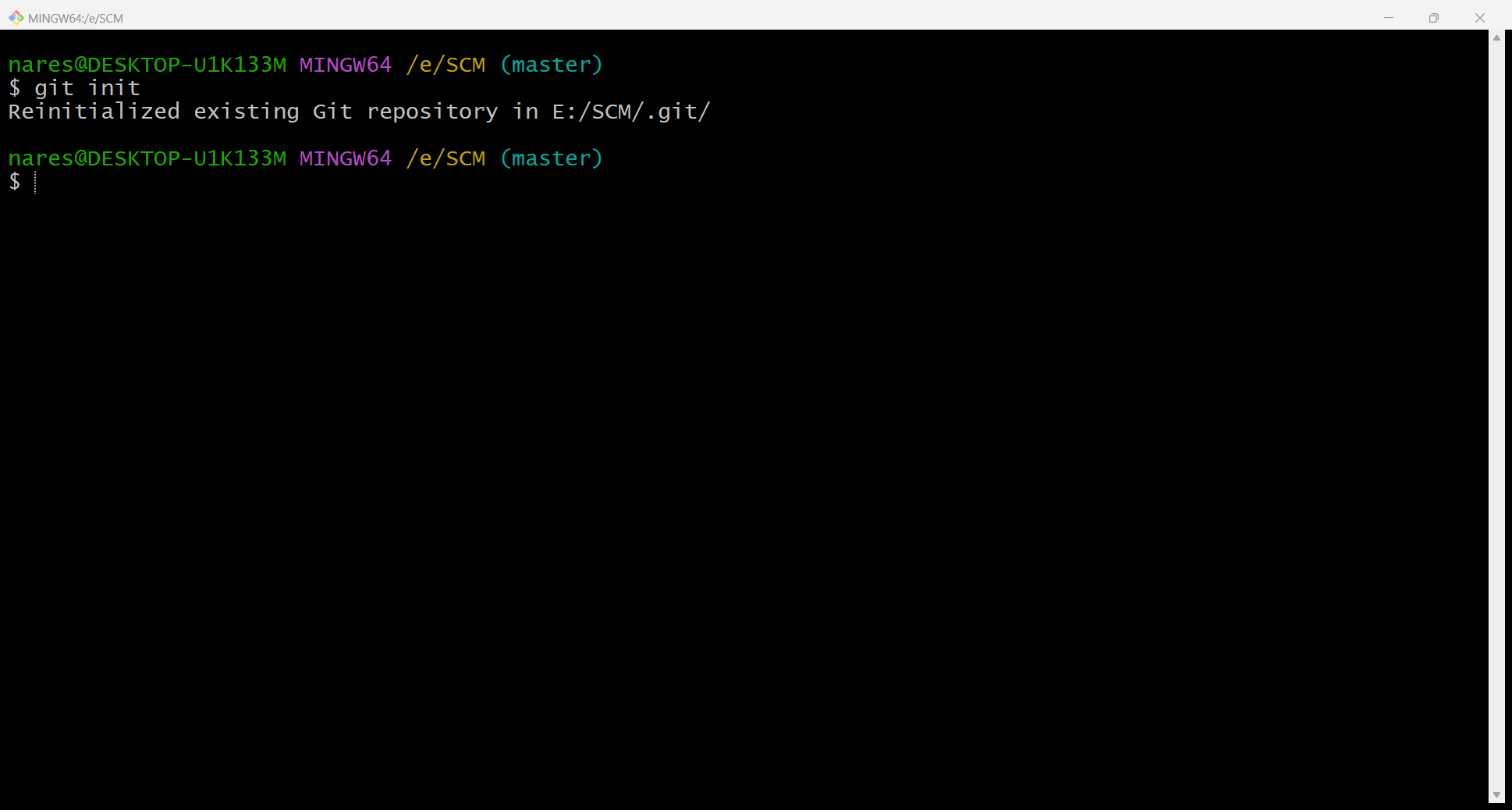
LOGS: logs are nothing but the history which we can see in git by using the code git log.

It contains all the past commits, insertions and deletions in it which we can see anytime. 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 “get init” and it creates a folder “. Git”.

Image:



When we use git for the first time, we have to give the username and email so that if I'm going to make some changes in project, it will be visible to all.

for this, we use command:

“git config--global user.name Name”

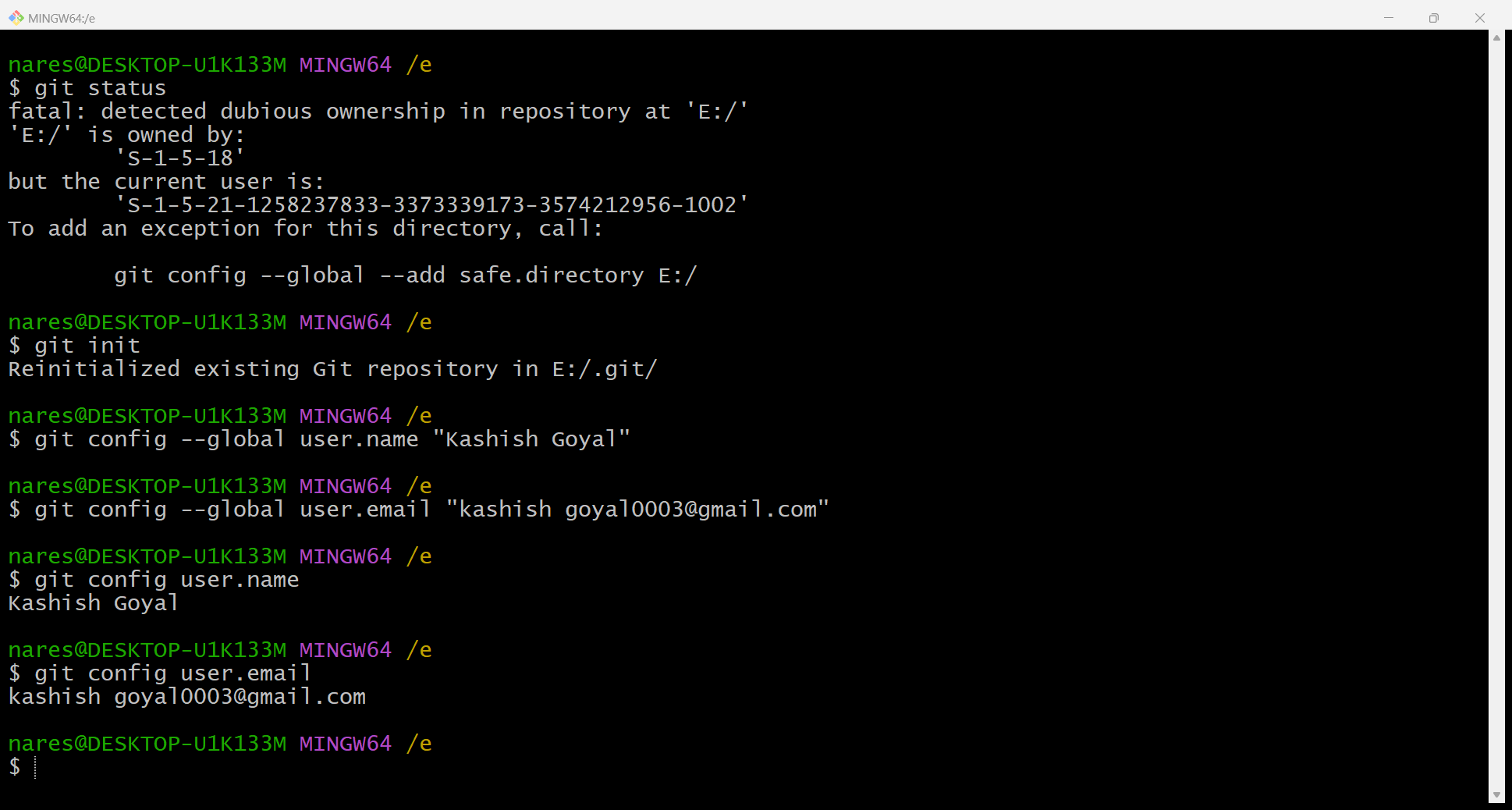
“git config--global user.email email “

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

“git config--global user.name”

“git config--global user.email”

IMAGES:

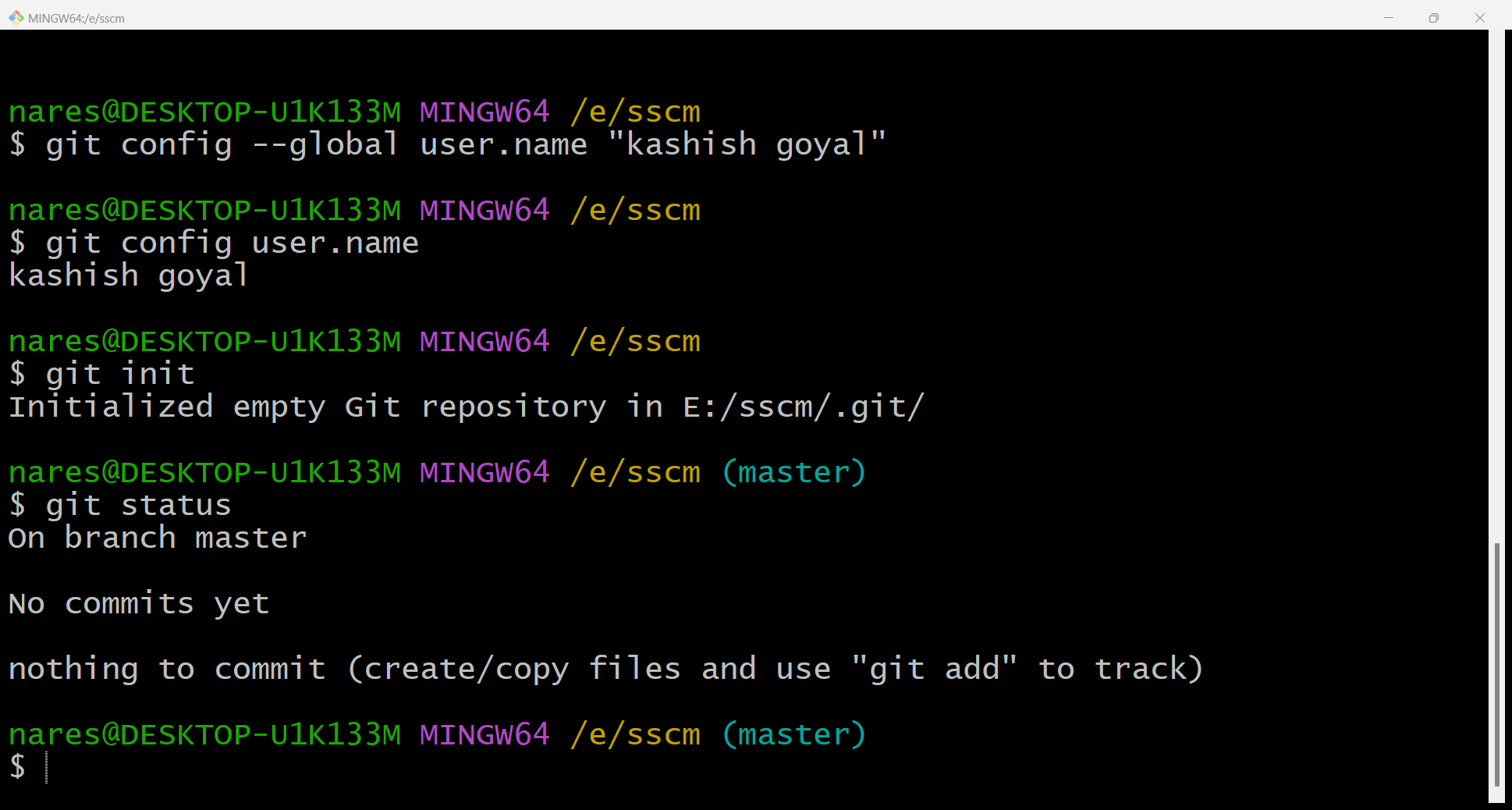


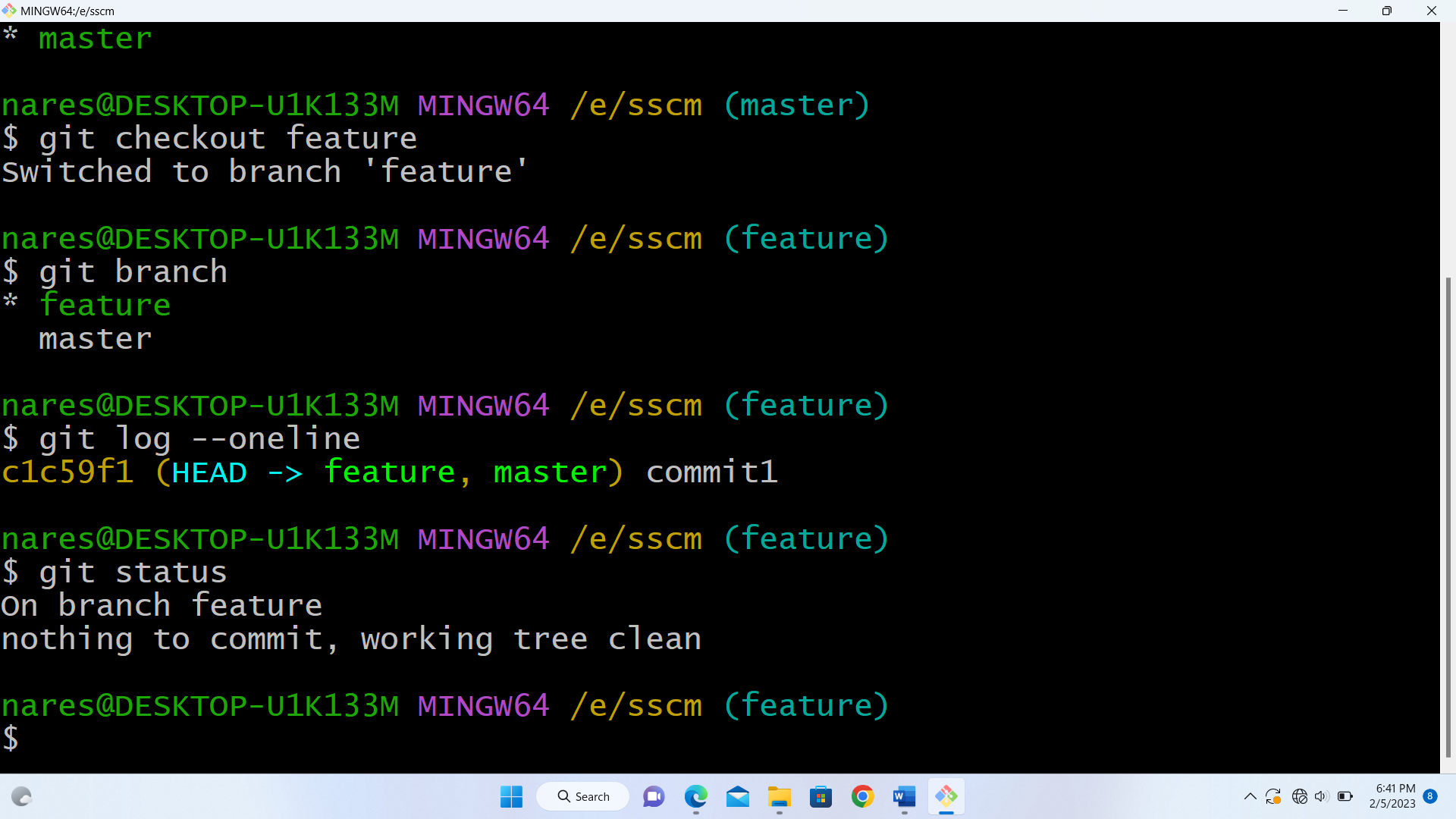
**SOME IMPORTANT COMMANDS**

* Is : it gives the file names in the folder.
* Is -lart : Gives the hidden files also.
* git status : display the state of the working directory add 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 the commits in a repository’s history.
* git diff : it compares my working tree to staging area.

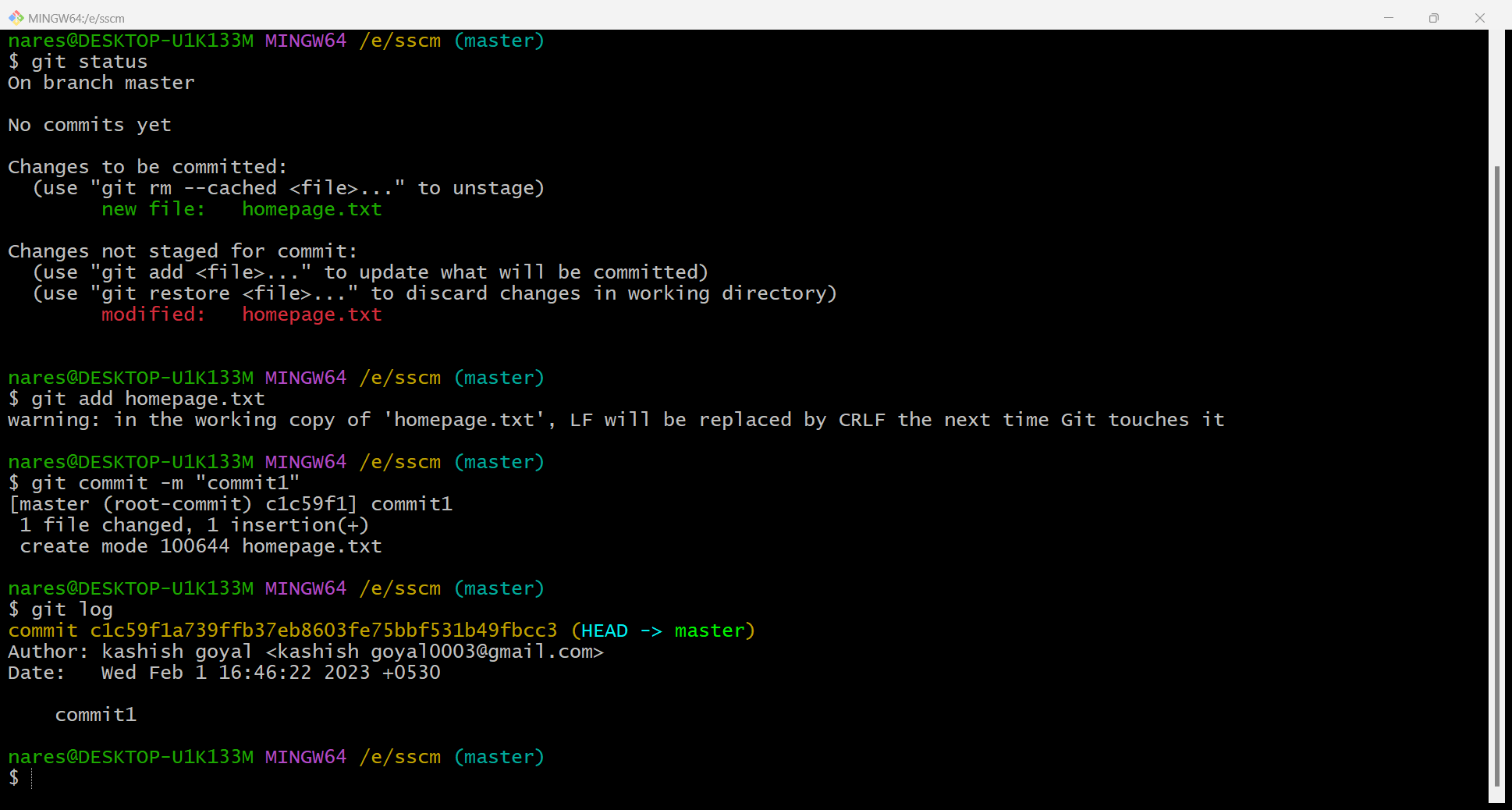
GIT STATUS:

IMAGE

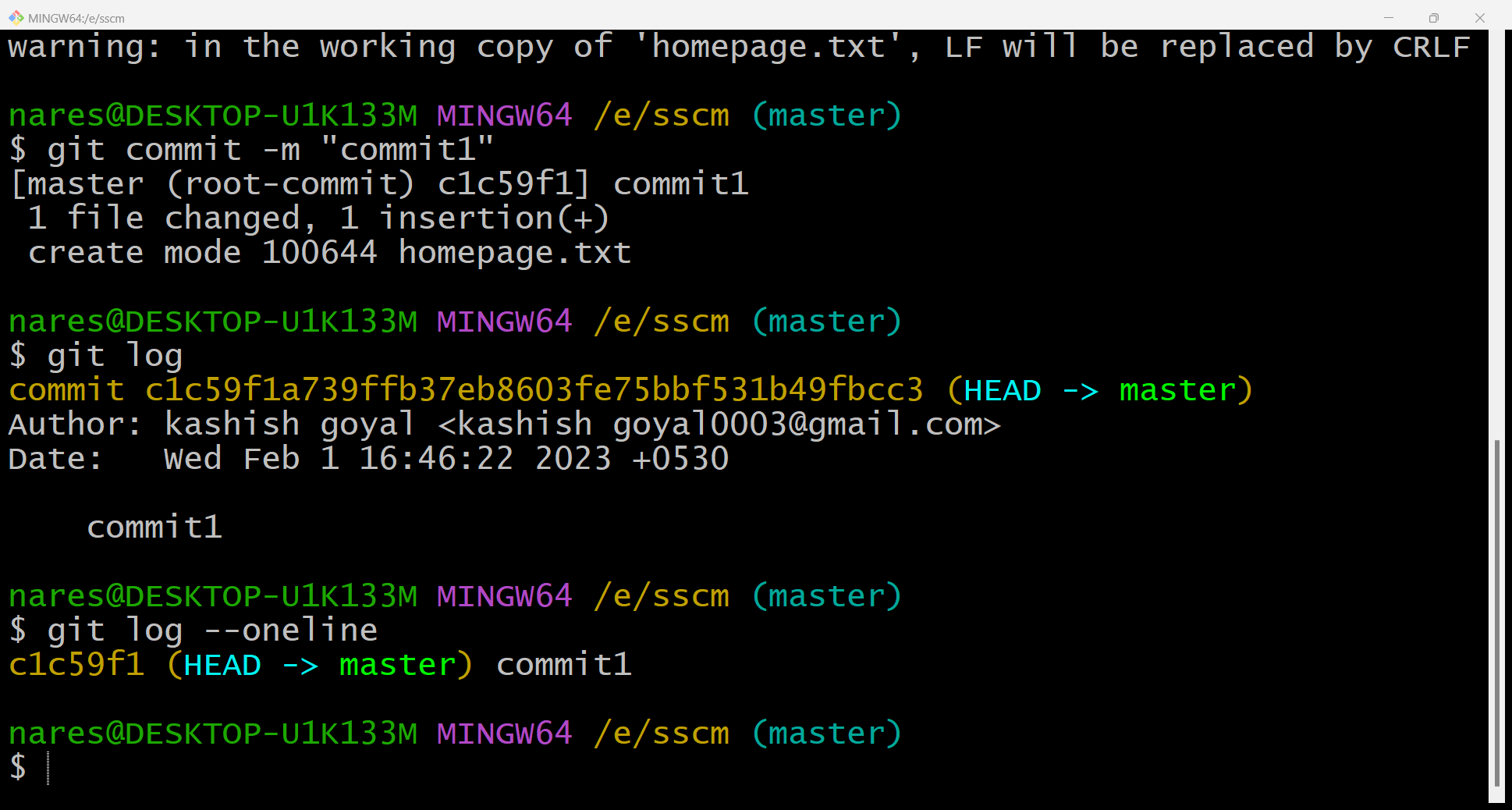




GIT LOG:



Git log in one line:



**EXPERIMENT 4**

AIM: create and visualise branches

THEORY:

Branching: A branch in GIT an independent line of work (a pointer to a specific commit). it allows user to create a branch from the original code (master branch).

Create branches: the main branch in git is call a master branch. all the files present in master can be shown in branch but the file which are created and branch are not shown in master branch. we can also merge both the parent (master) and child (other branches).

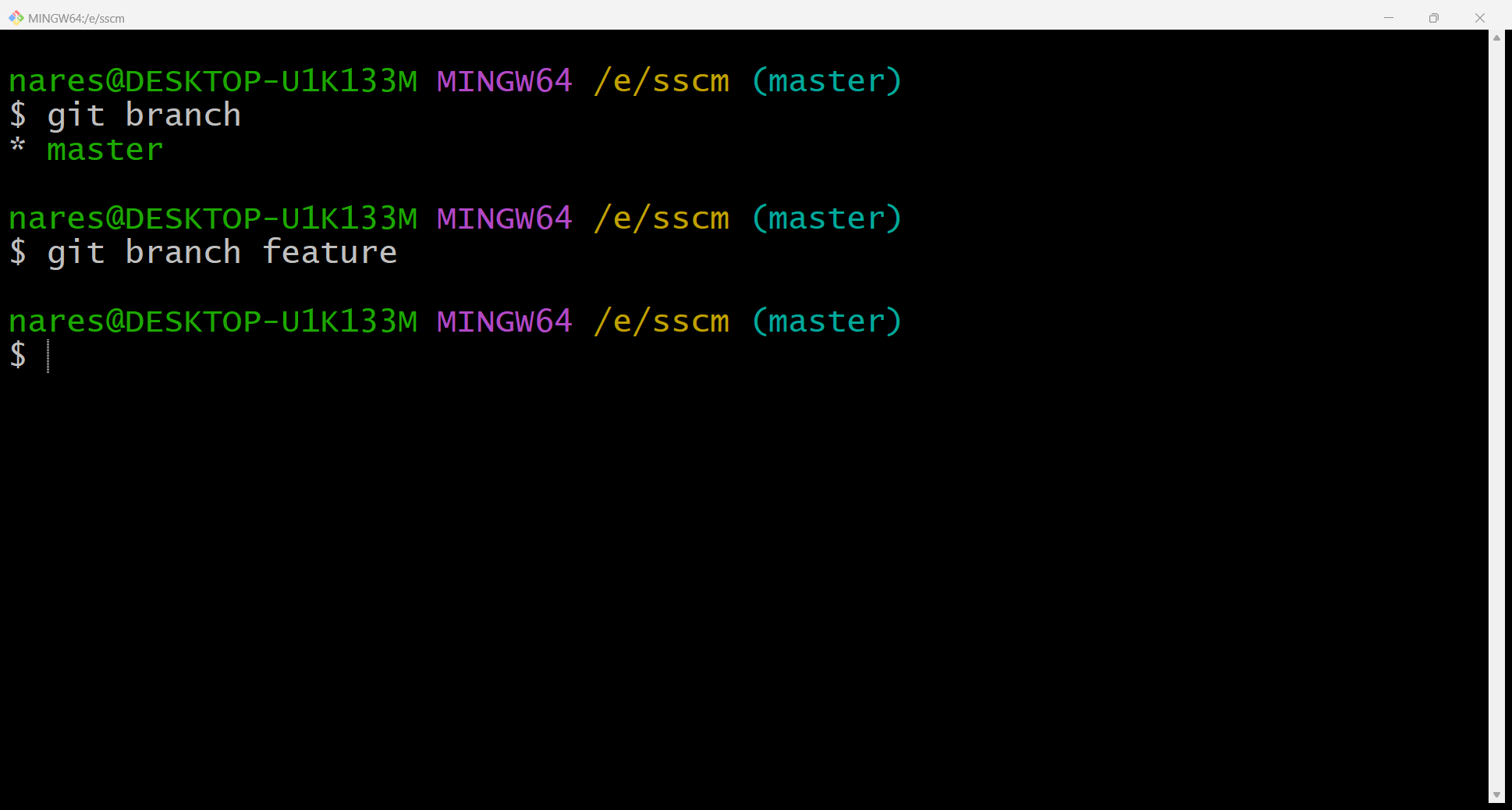
SYNTAX:

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

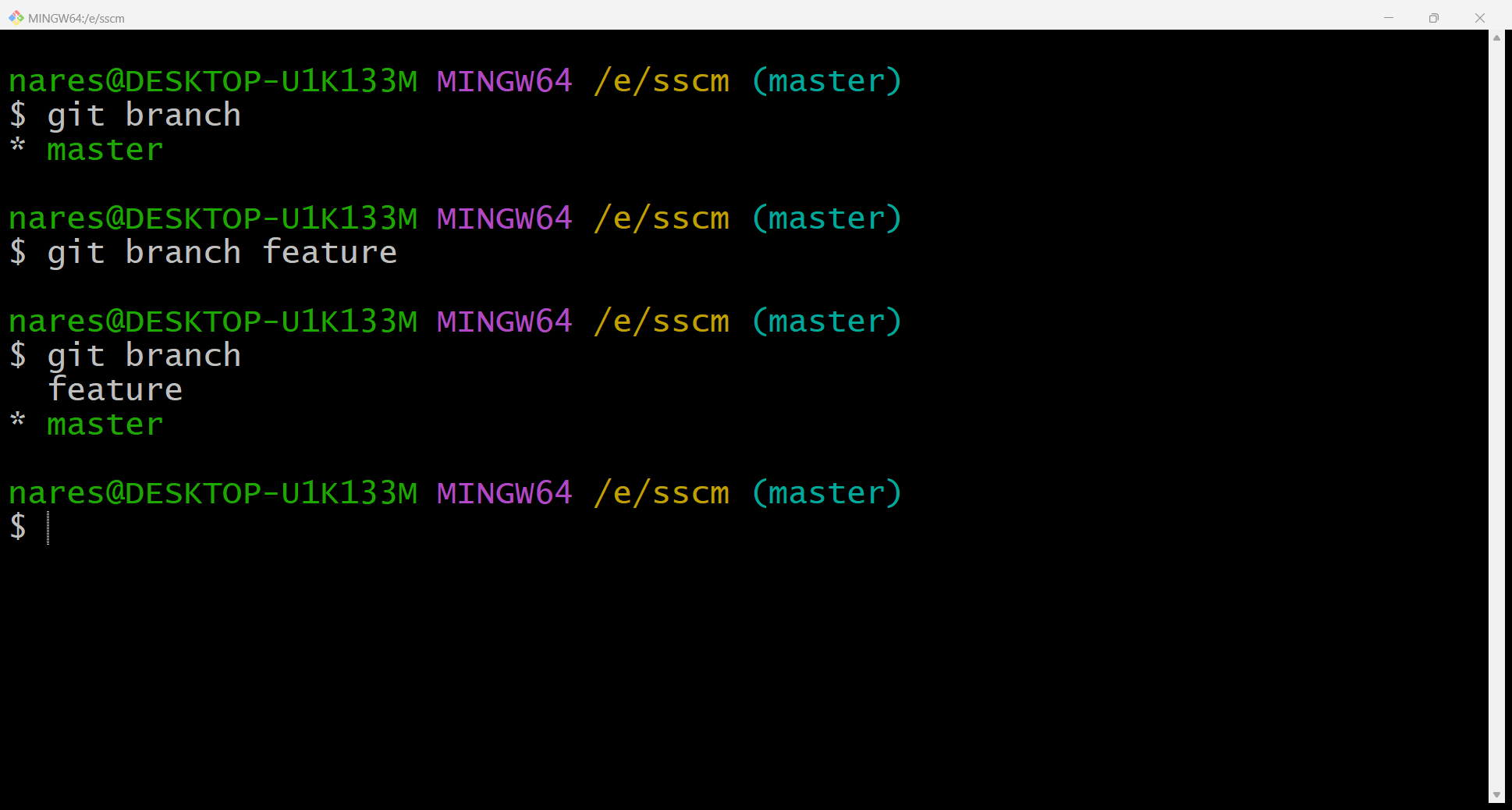


SNAPSHOTS:

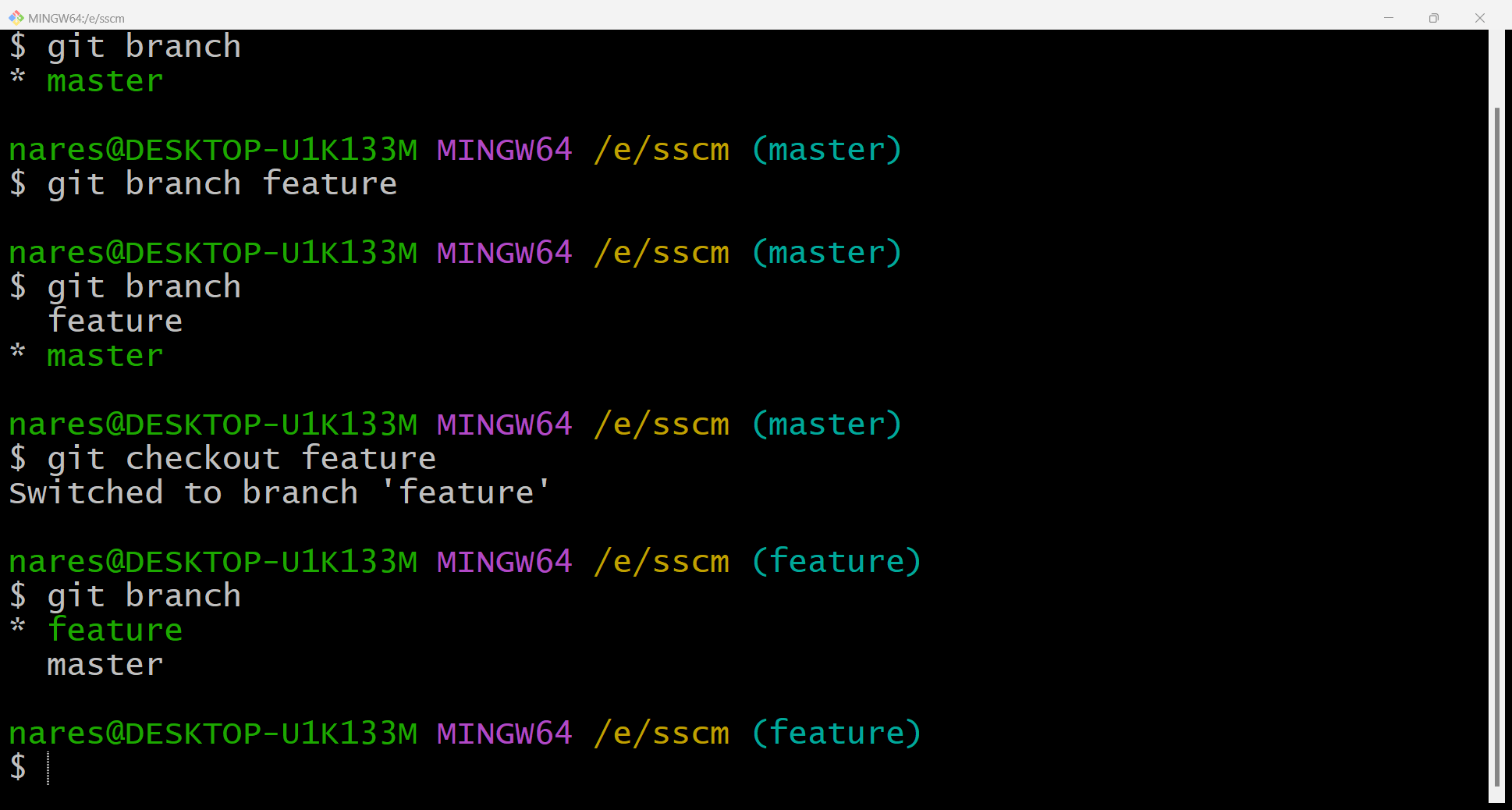
Adding feature branch



\* means that we are on master branch



Switching to feature branch



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

Timeline

Description automatically generated

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:

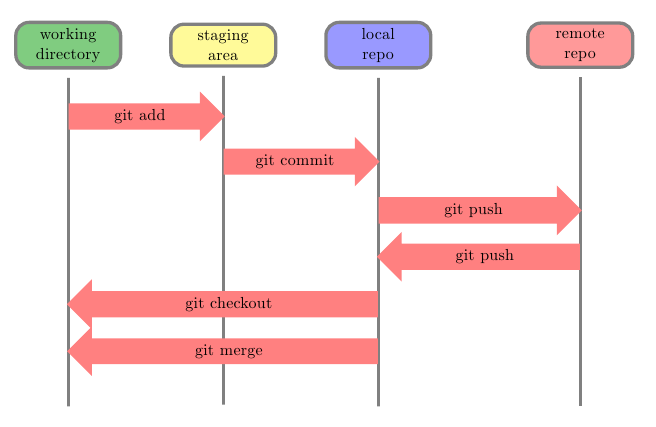
Staging area is the playground where you group, add and organise the files to be committed to git tracking their versions.

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 message, this step also records the author and the 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 git hub. And pushing means uploading the commits from local git repository to remote repository hosted in github.



**EXPERIMENT 6**

**Aim:** Add collaborators on GitHub Repo

**Theory:**

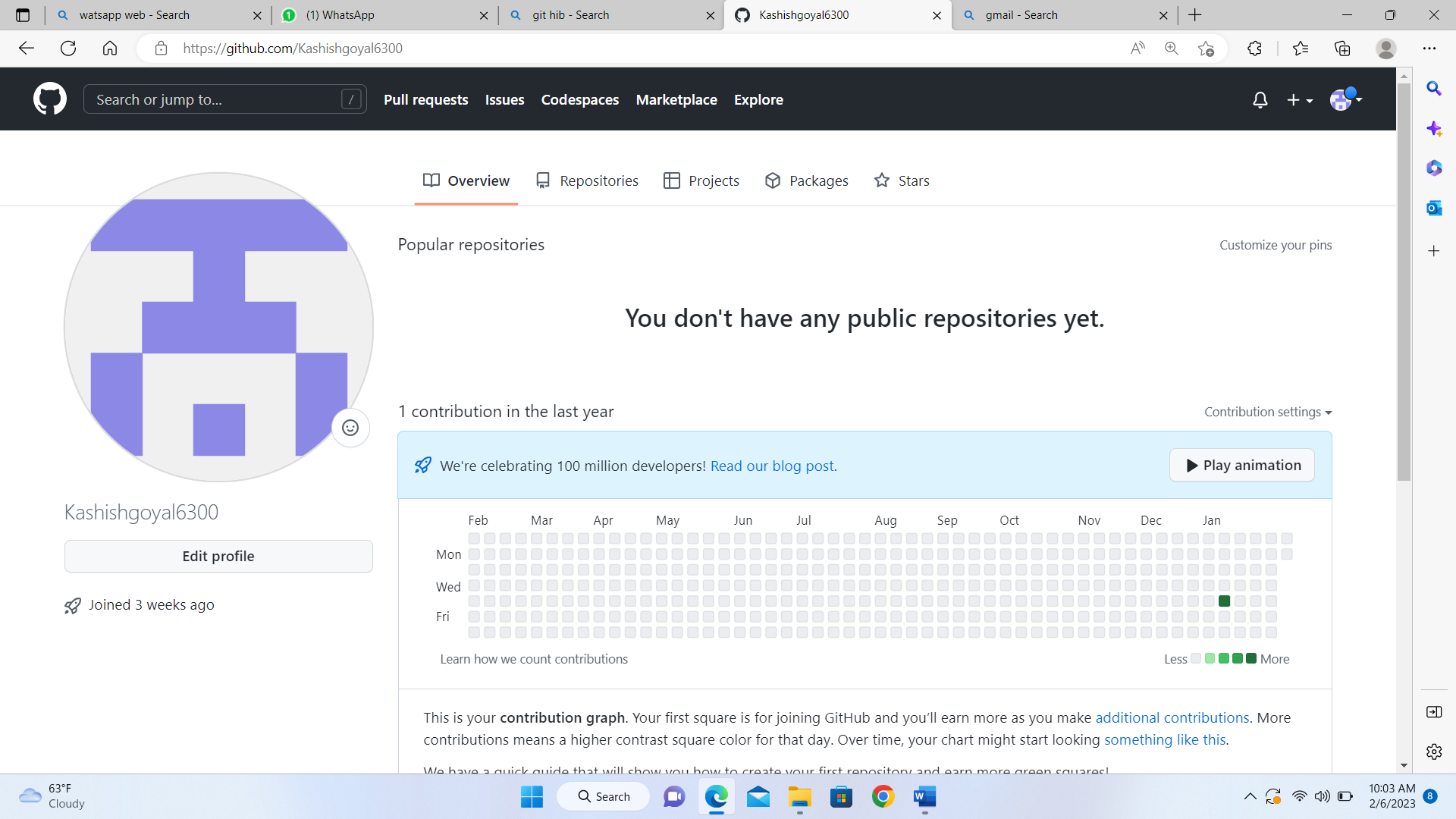
Whenever you make a repository in GitHub, not everyone has the permission to change or push codes into your repository. The users have a read-only access. In order to allow other individuals to make changes to your repository, you need to invite them to collaborate to the project.

GitHub also restricts the number of collaborators we can invite within a period of 24 hours. If we exceed the limit, then either we have to wait for 24-hours or we can also create an organization to collaborate with more people.

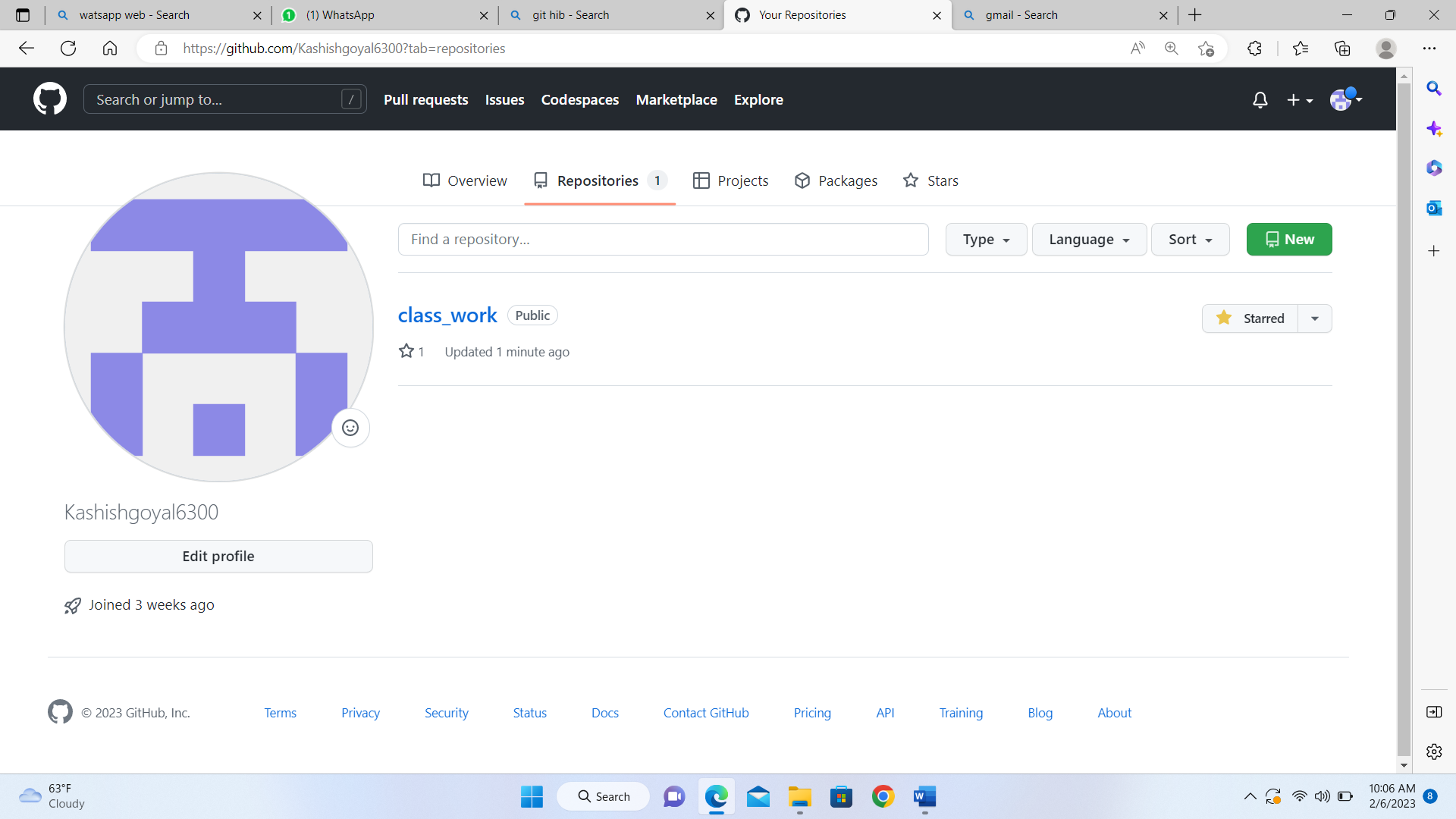
Being a collaborator, the user can create, merge and close pull requests in the repository. They can also remove them as the collaborator.

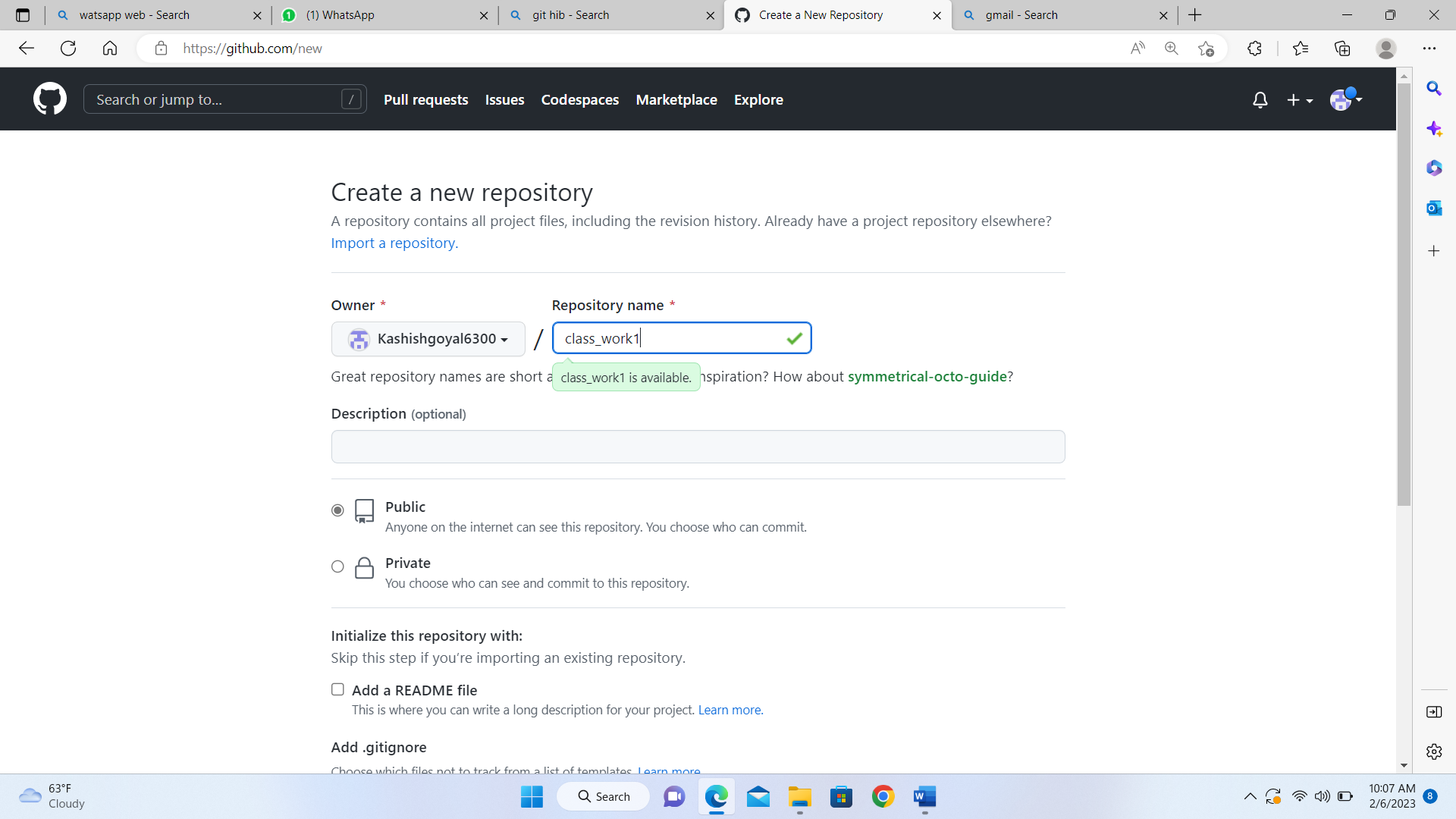
**Procedure:**

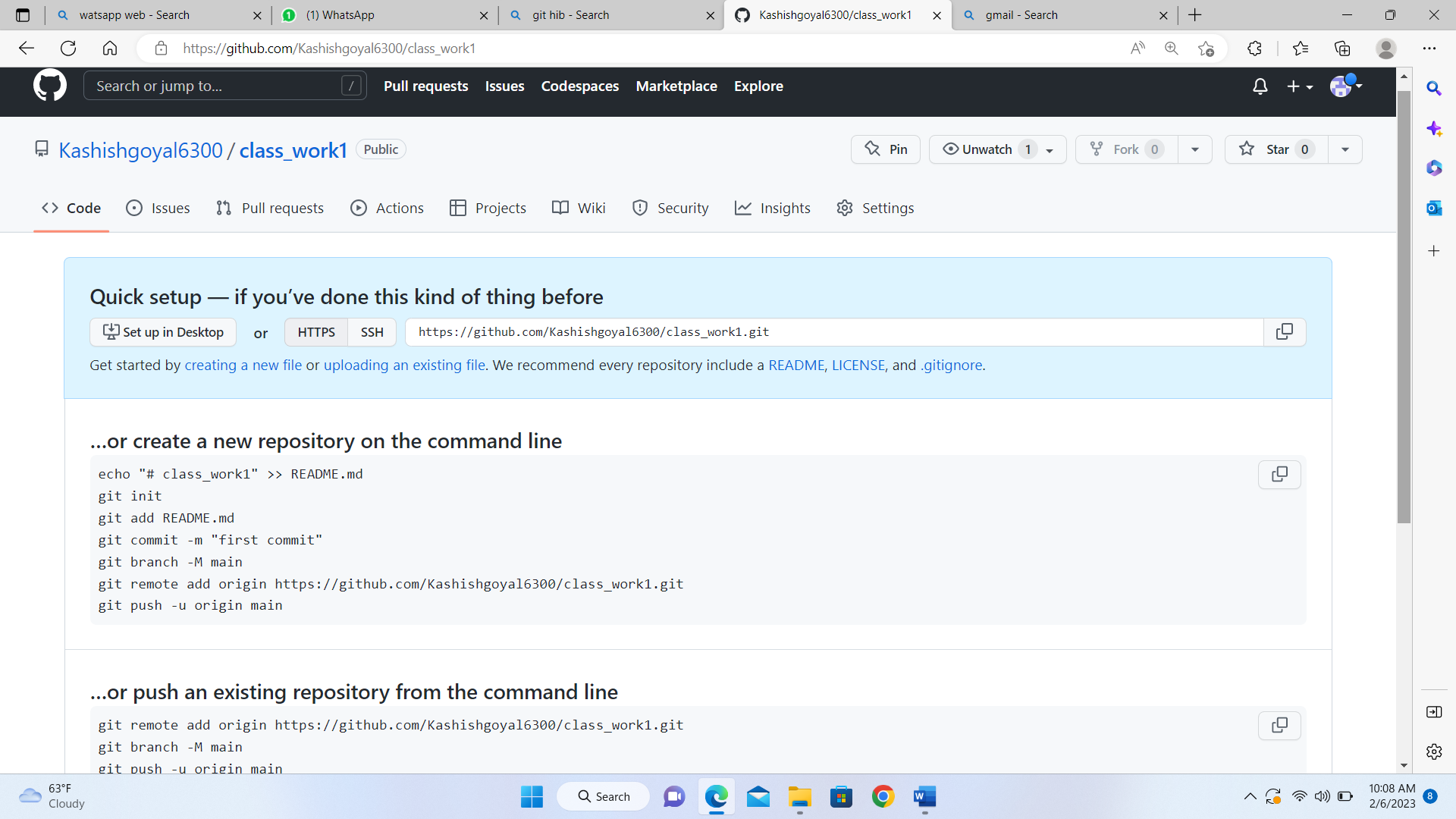
1. Login to your GitHub account and you will land on the homepage as shown below. Click on Repositories option in the menu bar.



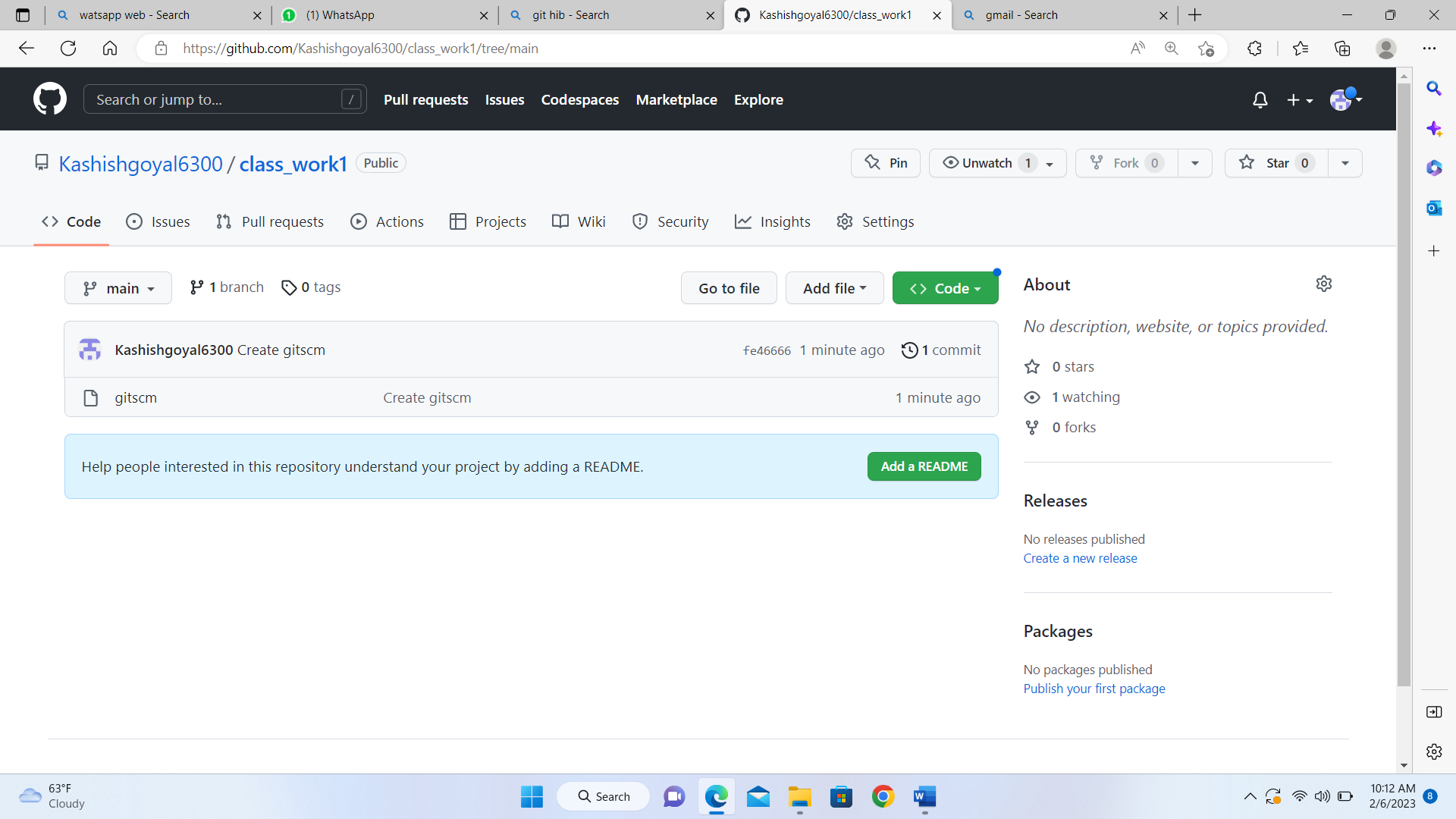
1. Click on the ‘New’ button in the top right corner.



1. Enter the Repository name and add the description of the repository.
2. if you want the repository to be public or private
3. 
4. If you want to import code from an existing repository select the import code option.

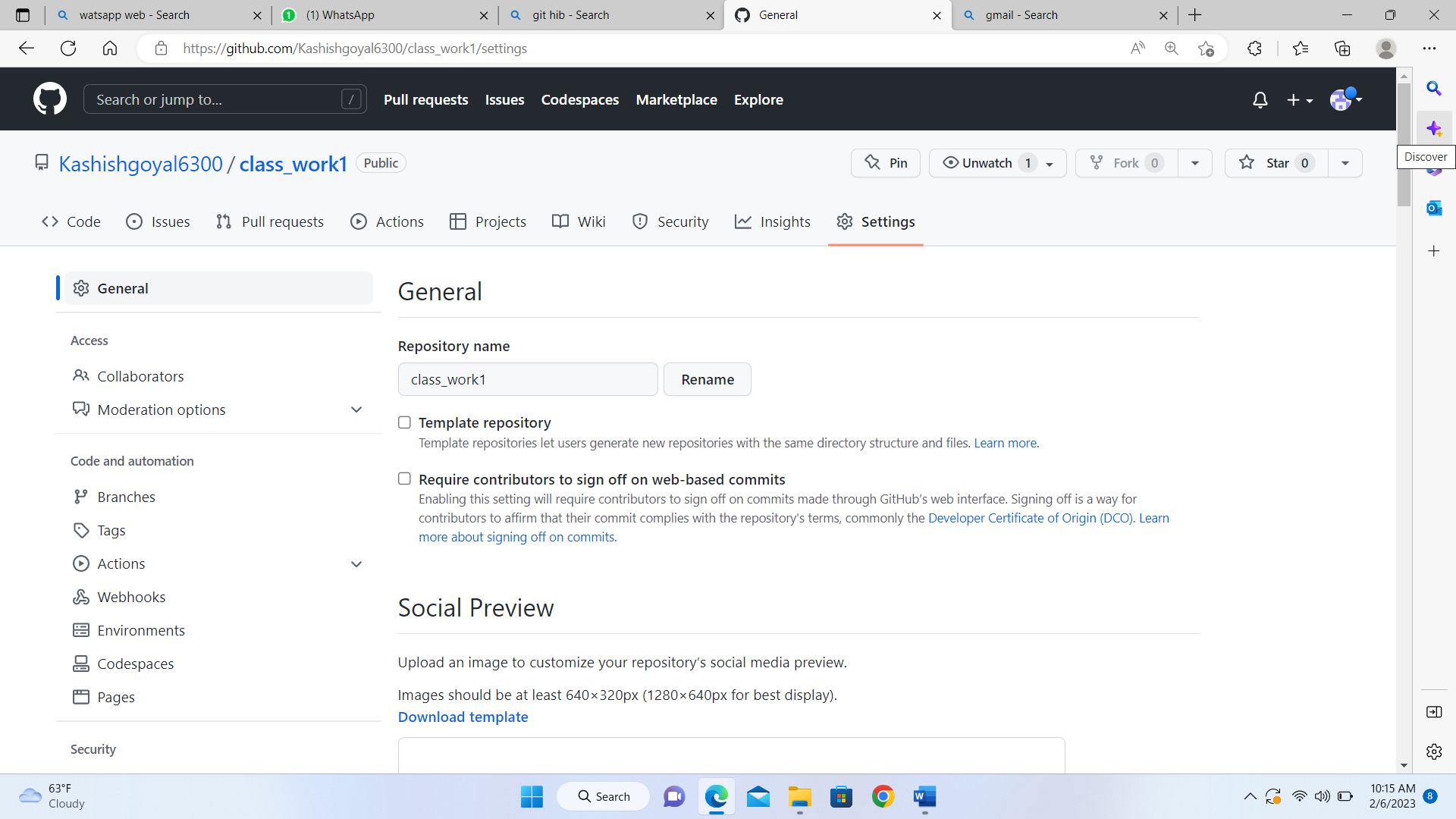


1. To add collaborators to your repository, open your



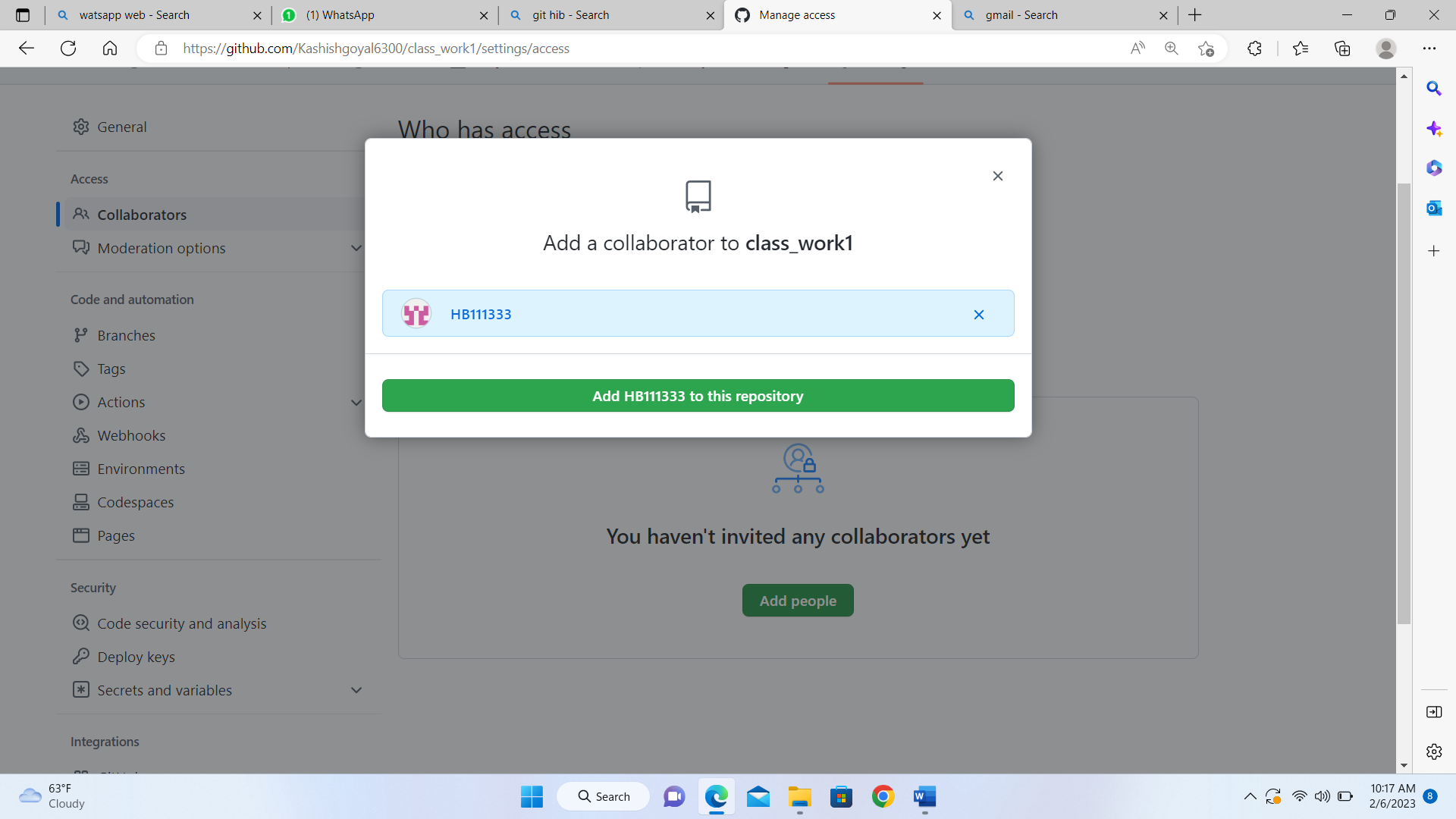
repository and select settings option in the navigation bar.

1. Click on Collaborators option

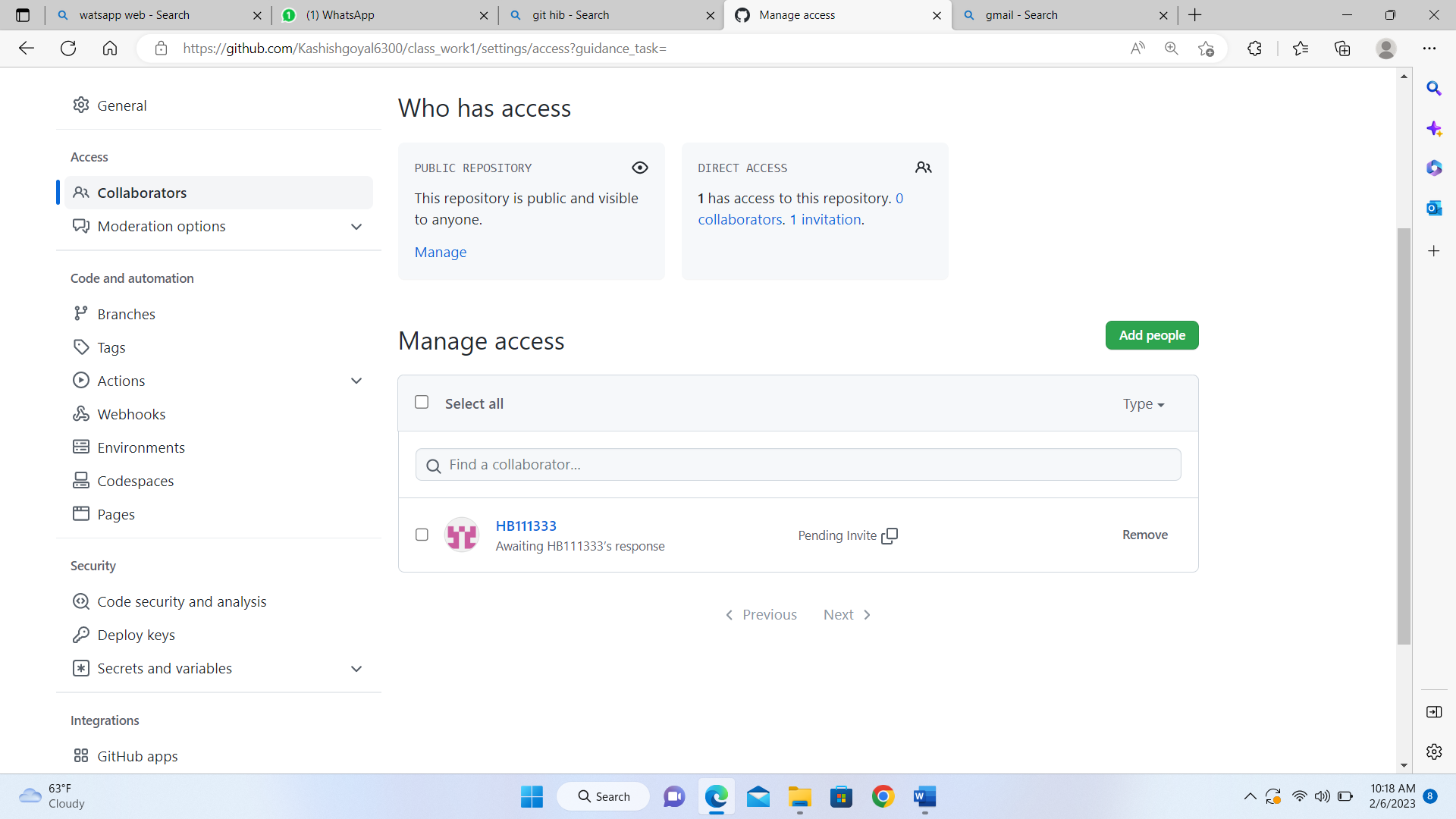


After clicking on collaborators, GitHub asks you to enter your password to confirm the access to the repository.

To add members, click on the add people option and search the id of your respective team member



To remove any member, click on remove option available in the last column of member’s respective row.

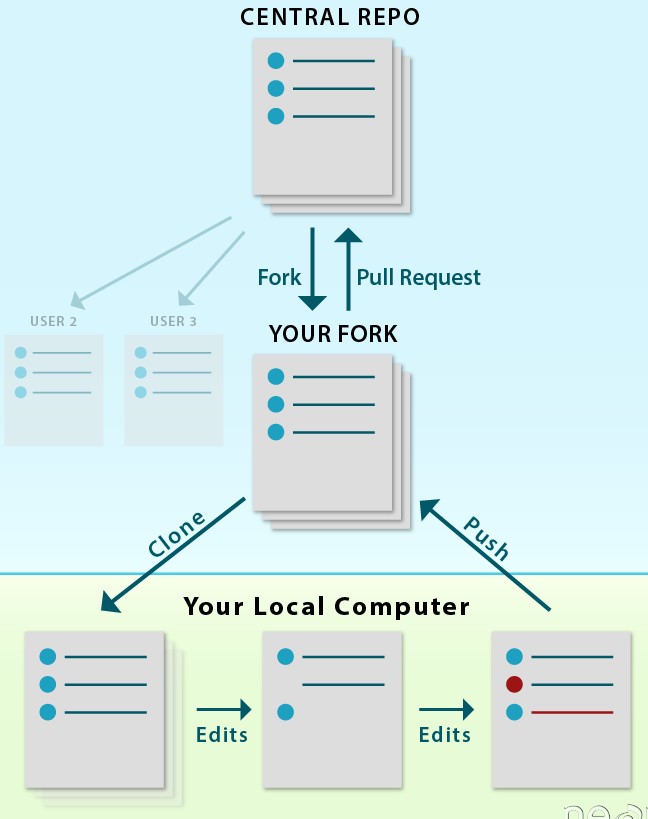




**EXPERIMENT 7**

## **Aim:** Fork and Commit

**Theory:** A fork is a copy of a repository that you manage. It allows us to freely experiment with the data. After creating a fork, we can make any desired change like adding collaborators, rename files, generate GitHub pages but all these changes won’t be reflected in the original repository.



reference for picture: https://www.earthdatascience.org

To import the changes into the original repository, the user needs to send a pull request to the maintainer. If the maintainer closes the pull request only then the content can be added to the original repository.

Forking is a better method than directly cloning any repository, as in cloning only the default branch is cloned whereas forking creates a clone of the complete repo and also allows us to push the changes to the main repository by using open and close pull request.

**Procedure:**

1. To fork a repository first thing you need to do is, sign in to your GitHub account and then you come to the repository you want to fork, so here for demo purpose am using **Group24-ChitkaraUniversity/2210990496** repository.