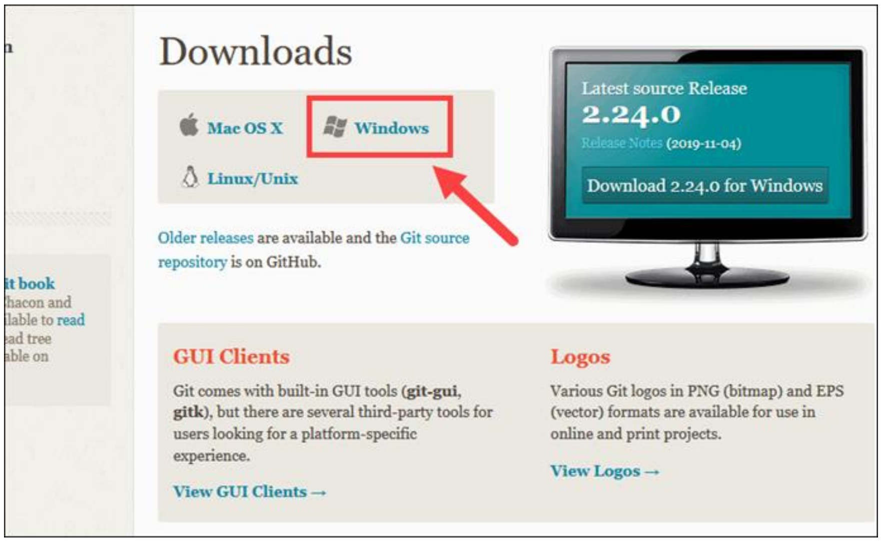
Experiment -1.1

Install Git and creating repository

Student Name:	SAKSHI	**UID:**	22BDO10064
Branch:	CSE(DEVOPS)	**Section/Group:**	22BCD-1/B
Semester:	3rd	**Date of Performance:**	
Subject Name:	Git and GitHub	**Subject Code:**	22CSH-293

1. **Aim/Overview of the practical:** Install Git and creating repository.
2. **Task to be done:** Download Git for Windows. And, to make repositories.
3. **Steps for experiment:** 1) Browse to the official Git website: https://git-scm.com/downloads. 2) Click the download link for Windows and allow the download to complete.


Downloads
Mac OS X
Windows
Linux/Unix
Older releases are available and the Git source repository is on GitHub.
Latest source Release
2.24.0
Release Notes (2019-11-04)
Download 2.24.0 for Windows
GUI Clients
Git comes with built-in GUI tools (git-gui, gitk), but there are several third-party tools for users looking for a platform-specific experience.
View GUI Clients →
Logos
Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.
View Logos →

3) Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.

4) Allow the app to make changes to your device by clicking **Yes** on the User Account Control dialog that opens.

5) Review the GNU General Public License, and when you're ready to install, click **Next**.

6) The installer will ask you for an installation location. Leave the default, unless you have reason to change it, and click **Next**.

7) A component selection screen will appear. Leave the defaults unless you have a specific need to change them and click **Next**.

8) The installer will offer to create a start menu folder. Simply click **Next**.

9) Select a text editor you'd like to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click **Next**.

10) The next step allows you to choose a different name for your initial branch. The default is 'master.' Unless you're working in a team that requires a different name, leave the default option and click **Next.**

11) This installation step allows you to change the **PATH environment**. The **PATH** is the default set of directories included when you run a command from the command line. Leave this on the middle (recommended) selection and click **Next**.

12) The installer now asks which SSH client you want Git to use. Git already comes with its own SSH client, so if you don't need a specific one, leave the default option and click **Next.**

13) The next option relates to server certificates. Most users should use the default. If you're working in an Active Directory environment, you may need to switch to Windows Store certificates. Click **Next**.

14) The next selection converts line endings. It is recommended that you leave the default selection. This relates to the way data is formatted and changing this option may cause problems. Click **Next**.

15) Choose the terminal emulator you want to use. The default MinTTY is recommended, for its features. Click **Next**.

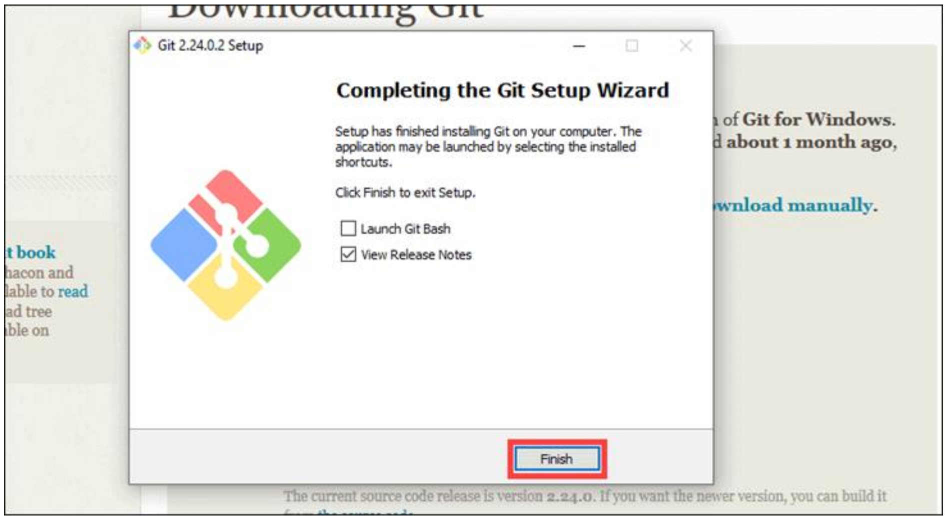
16) The installer now asks what the **git pull** command should do. The default option is recommended unless you specifically need to change its behavior. Click **Next** to continue with the installation.

17) Next you should choose which credential helper to use. Git uses credential helpers to fetch or save credentials. Leave the default option as it is the most stable one, and click **Next**.

18) The default options are recommended, however this step allows you to decide which extra option you would like to enable. If you use symbolic links, which are like shortcuts for the command line, tick the box. Click **Next**.

19) Depending on the version of Git you're installing, it may offer to install experimental features. At the time this article was written, the options to include support for pseudo controls and a built-in file system monitor were offered. Unless you are feeling adventurous, leave them unchecked and click **Install**.

20) Once the installation is complete, tick the boxes to view the Release Notes or Launch Git Bash, then click **Finish**.



Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

How to Launch Git in Windows

Git has two modes of use – a **bash scripting shell** (or command line) and a **graphical user interface** (GUI).

To launch **Git GUI** open the **Windows Start** menu, type ***git gui*** and press **Enter** (or click the application icon).

Connecting to a Remote Repository

You need a GitHub username and password for this next step.

Create a Test Directory

Open a Windows PowerShell interface by pressing **Windows Key + x**, and then **i** once the menu appears.

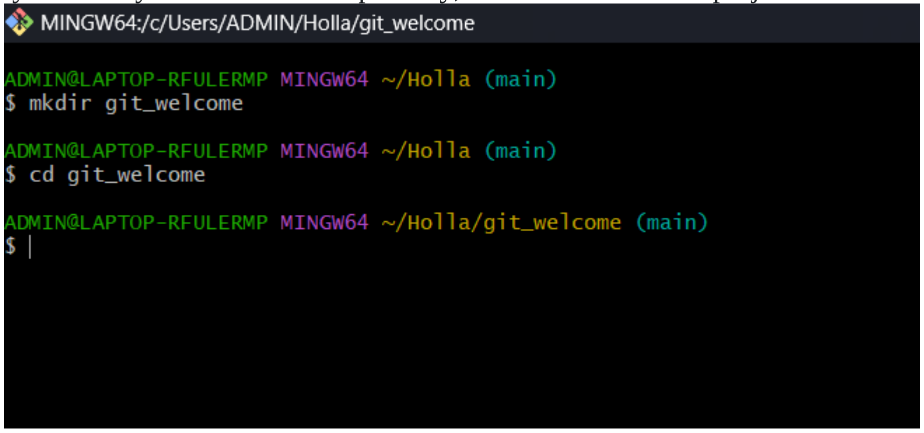
Create a new test directory (folder) by entering the following:

```
mkdir git_test
```

Change your location to the newly created directory:

```
cd git_test
```

Note: If you already have a GitHub repository, use the name of that project instead of **git_test**



```
MINGW64:/c/Users/ADMIN/Holla/git_welcome

ADMIN@LAPTOP-RFULERMP MINGW64 ~/Holla (main)
$ mkdir git_welcome

ADMIN@LAPTOP-RFULERMP MINGW64 ~/Holla (main)
$ cd git_welcome

ADMIN@LAPTOP-RFULERMP MINGW64 ~/Holla/git_welcome (main)
$ |
```

Configure GitHub Credentials

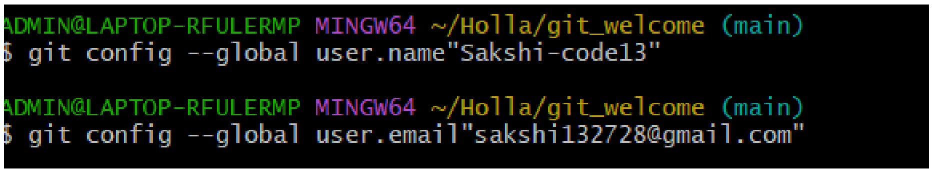
Configure your local Git installation to use your GitHub credentials by entering the following:

git config --global user.name "github_username"

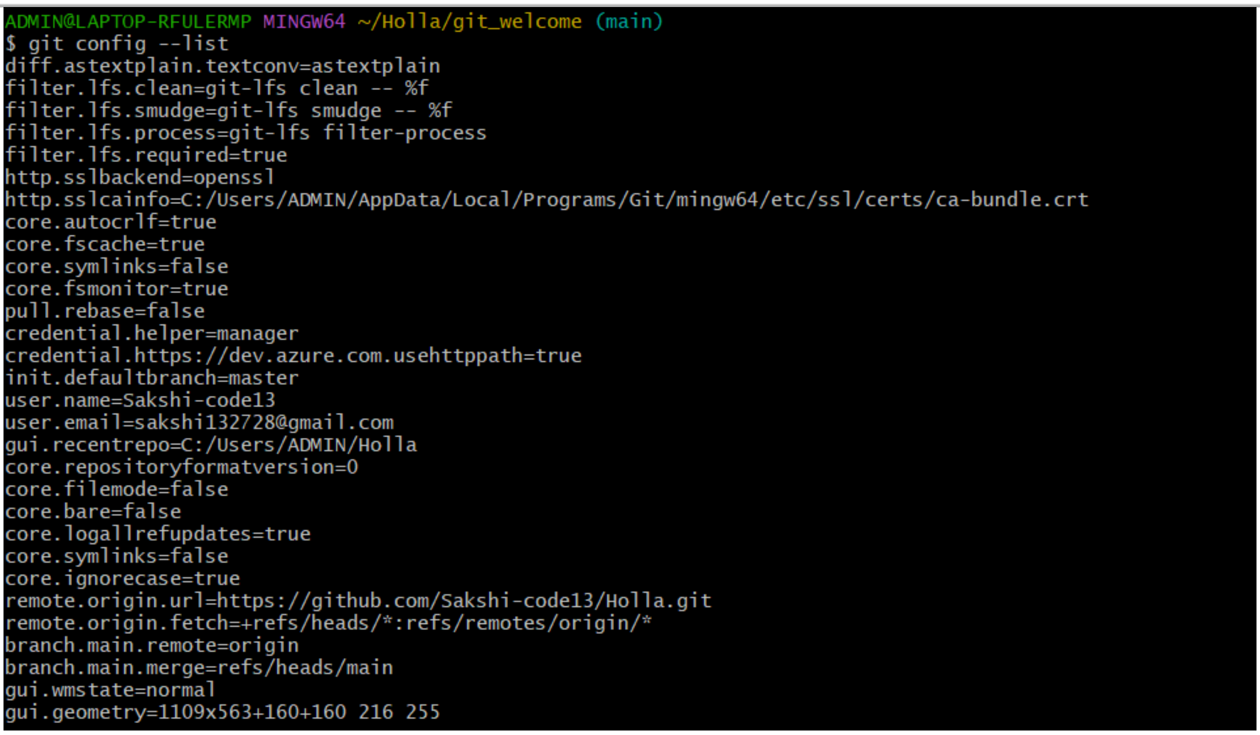
git config --global user.email "email_address"

Note: Replace **github_username** and **email_address** with your GitHub credentials.

We can also see the list of configurations by using the command git config – list.



```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Holla/git_welcome (main)
$ git config --global user.name"Sakshi-code13"

ADMIN@LAPTOP-RFULERMP MINGW64 ~/Holla/git_welcome (main)
$ git config --global user.email"sakshi132728@gmail.com"
```



```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Holla/git_welcome (main)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Users/ADMIN/AppData/Local/Programs/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.fsmonitor=true
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Sakshi-code13
user.email=sakshi132728@gmail.com
gui.recentrepo=C:/Users/ADMIN/Holla
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/Sakshi-code13/Holla.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
gui.wmstate=normal
gui.geometry=1109x563+160+160 216 255
```

Clone a GitHub Repository

Go to your repository on GitHub. In the top right above the list of files, open the **Clone or download** drop-down menu. Copy the **URL for cloning over HTTPS**.

Switch to your PowerShell window, and enter the following:

git clone repository_url

List Remote Repositories

Your working directory should now have a copy of the repository from GitHub. It should contain a directory with the name of the project. Change to the directory:

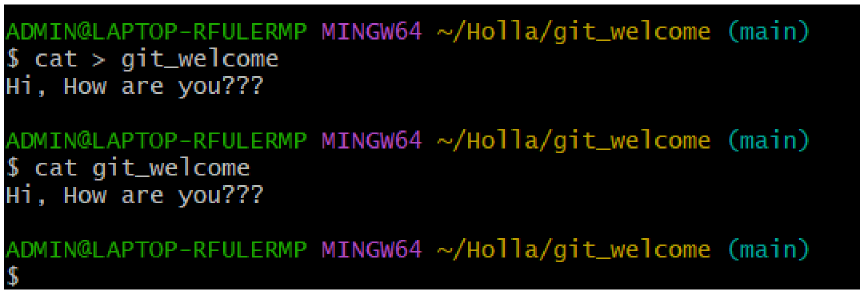
cd git_project

Once you’re in the sub-directory, list the remote repositories:

git remote -v

Enter the details in the current file
You can enter the desired details into your current file by the following commands:

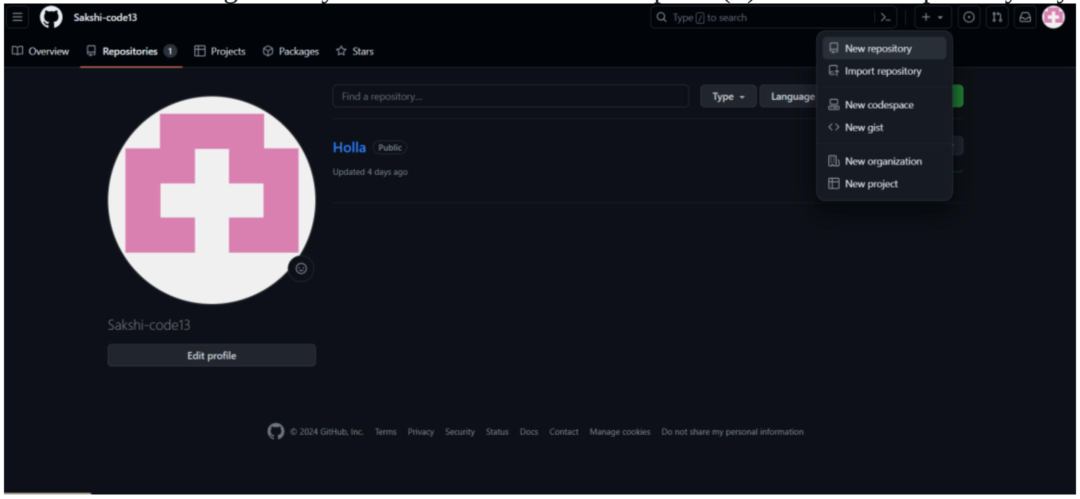
Cat > git_welcome
Enter the details in the file
Press Cntrl+D
Then, you can see the entered details through the command :- Cat git_welcome



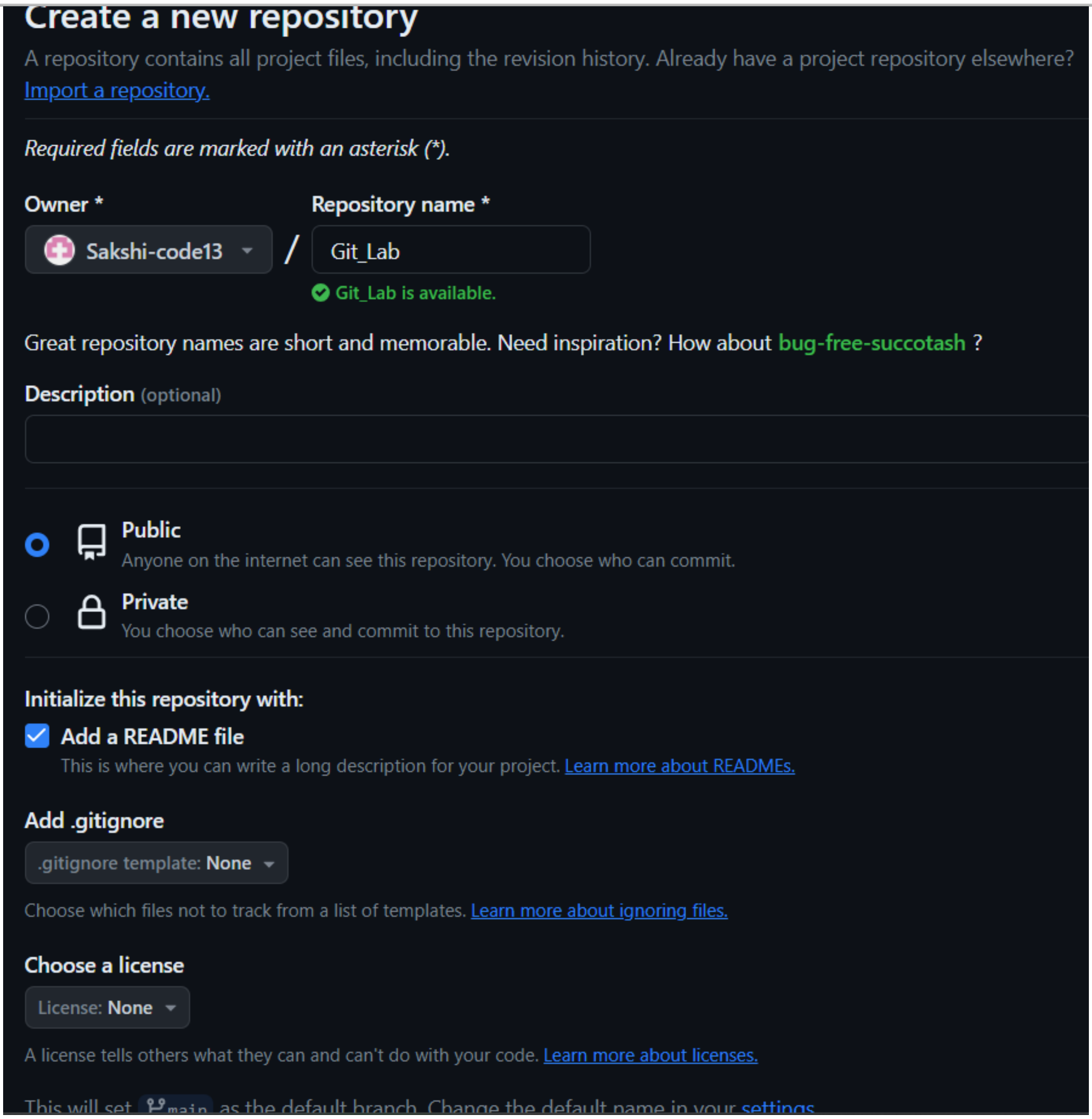
```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Holla/git_welcome (main)
$ cat > git_welcome
Hi, How are you???

ADMIN@LAPTOP-RFULERMP MINGW64 ~/Holla/git_welcome (main)
$ cat git_welcome
Hi, How are you???

ADMIN@LAPTOP-RFULERMP MINGW64 ~/Holla/git_welcome (main)
$
```

Creating Repository on GitHub

1. After successful login into your account. Click on the option (+) to add new repository to your account.



2. After clicking **new repository** option, we will have to initialize some things like, **naming our project**, choosing the **visibility** etc. After performing these steps click **Create Repository** button.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner *: Sakshi-code13 / Repository name *: Git_Lab

✓ Git_Lab is available.

Great repository names are short and memorable. Need inspiration? How about bug-free-succotash ?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☑ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

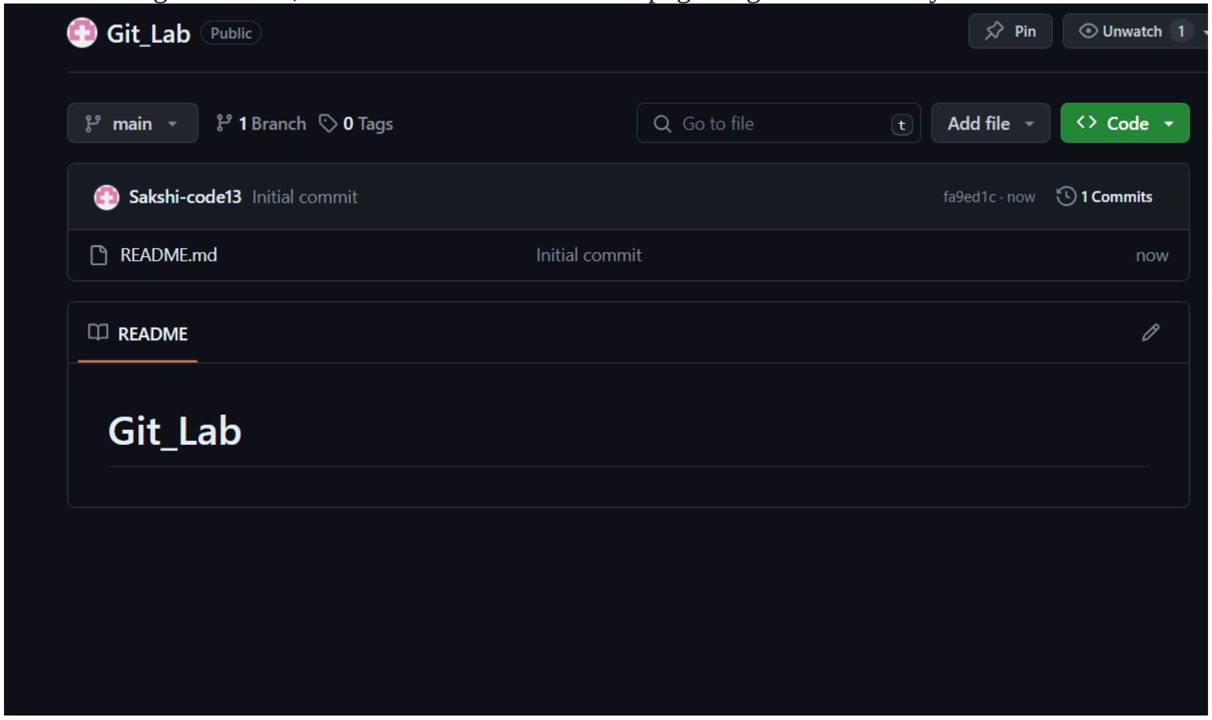
Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

This will set main as the default branch. Change the default name in your settings.

3. After clicking the button, we will be directed to below page. Right now the only file we have is a readme file.



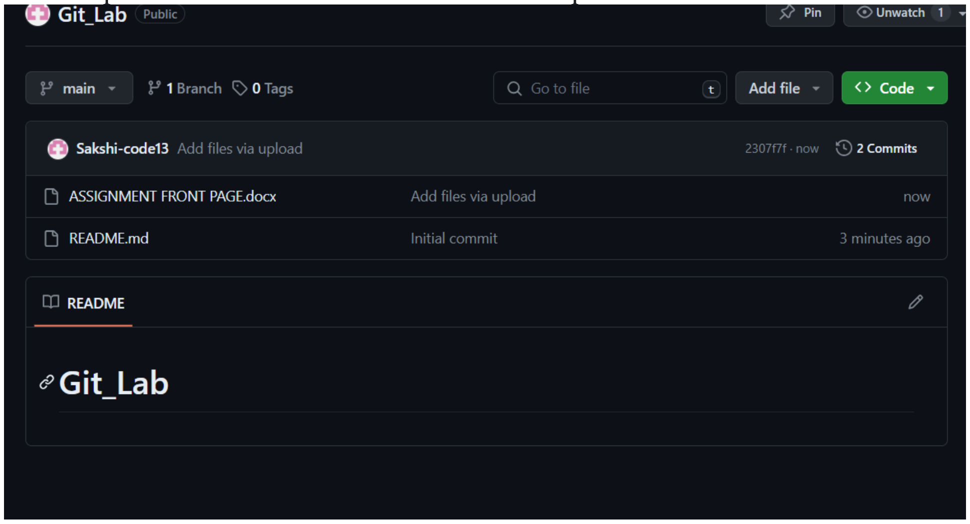
Sakshi-code13 Initial commit fa9ed1c · now 1 Commits

README.md	Initial commit	now

README

Git_Lab

4. Now click on the "Upload files" button. Do some needed steps. Now we can see all the files in our github.



Sakshi-code13 Add files via upload 2307f7f · now 2 Commits

ASSIGNMENT FRONT PAGE.docx	Add files via upload	now
README.md	Initial commit	3 minutes ago

README

Git_Lab

4. Result/Output/Writing Summary:

We have successfully created a repository and applied some commands on that.

Learning outcomes (What I have learnt):

1. Learnt about GitHub.

2. Learnt about Git.

3. Learnt about various git commands that can be applied on Git Bash.

4. Learnt about repositories.

5. Learnt about how to pull request and push.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			