

Contents

1. Introduction	1
2. Some of the common terms need to understand when using GitHub	2
3. Git Bash	3
4. Creating & uploading a project on GitHub using Git Terminal	6
5. Creating & uploading a project on GitHub using Git GUI	13
6. Conclusion	28

1. Introduction

Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

It is used for:

- Tracking code changes
- Tracking who made changes
- Coding collaboration

It is a web-based interface that uses Git, the open source version control software that lets multiple people makes separate changes to web pages at the same time. It allows for real-time collaboration, GitHub encourages teams to work together to build and edit their site content.

GitHub allows multiple developers to work on a single project at the same time, reduces the risk of duplicative or conflicting work, and can help decrease production time. With GitHub, developers can build code, track changes and innovate solutions to problems that might arise during the site development process simultaneously. Non-developers can also use it to create, edit, and update website content.

2. Some of the common terms need to understand when using GitHub.

- ❖ **Repository (repo)** - A folder in which all files and their version histories are stored.
- ❖ **Branch** - A workspace in which you can make changes that won't affect the live site.
- ❖ **Commit Changes** - A saved record of a change made to a file within the repo.
- ❖ **Pull Request (PR)** - The way to ask for changes made to a branch to be merged into another branch that also allows for multiple users to see, discuss and review work being done.
- ❖ **Merge** - After a pull request is approved, the commit will be pulled in (or merged) from one branch to another and then, deployed on the live site.
- ❖ **Issues** - How work is tracked when using git. Issues allow users to report new tasks and content fixes, as well as allows users to track progress on a project board from beginning to end of a specific project.
- ❖ **Federalist** - A platform that securely deploys a website from a GitHub repository in minutes and lets users preview proposed and published changes.
- ❖ **Markdown (.md)** - A way to write in Github that converts plain text to GitHub code. Sites such as Atom and Sublime Text are examples of free resources for developers using Markdown.
- ❖ **Fork** - A fork is a copy of a repository. This is useful when you want to contribute to someone else's project or start your own project based on theirs.
- ❖ **Clone** - A clone is a full copy of a repository, including all logging and versions of files.

3. Git Bash

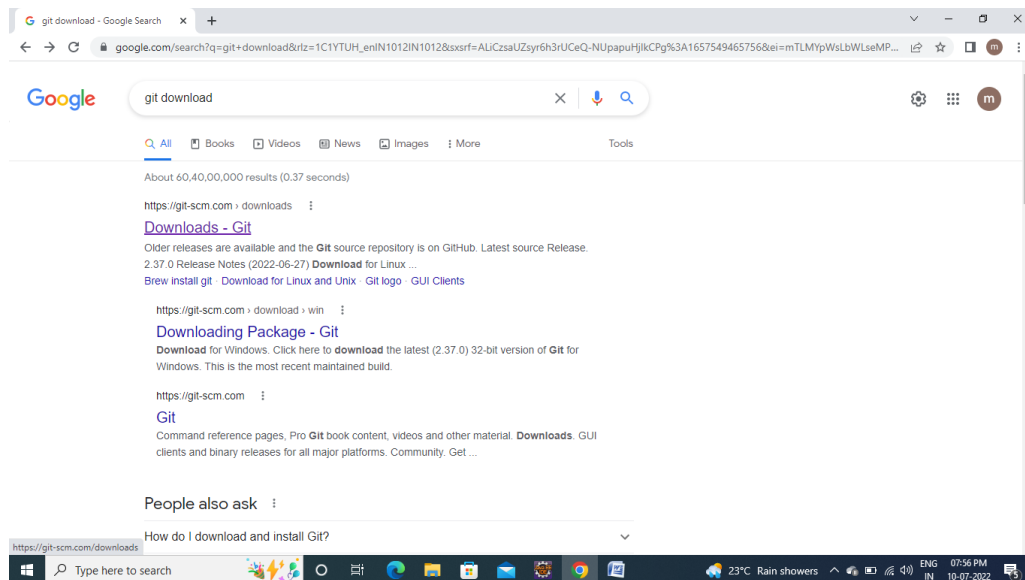
Git Bash is an application that provides Git command line experience on the Operating System. It is a command-line shell for enabling git with the command line in the system. A shell is a terminal application used to interface with an operating system through written commands. Git Bash is a package that installs Bash, some common bash utilities, and Git on a Windows operating system. In Git Bash the user interacts with the repository and git elements through the commands.

Installing Git Bash

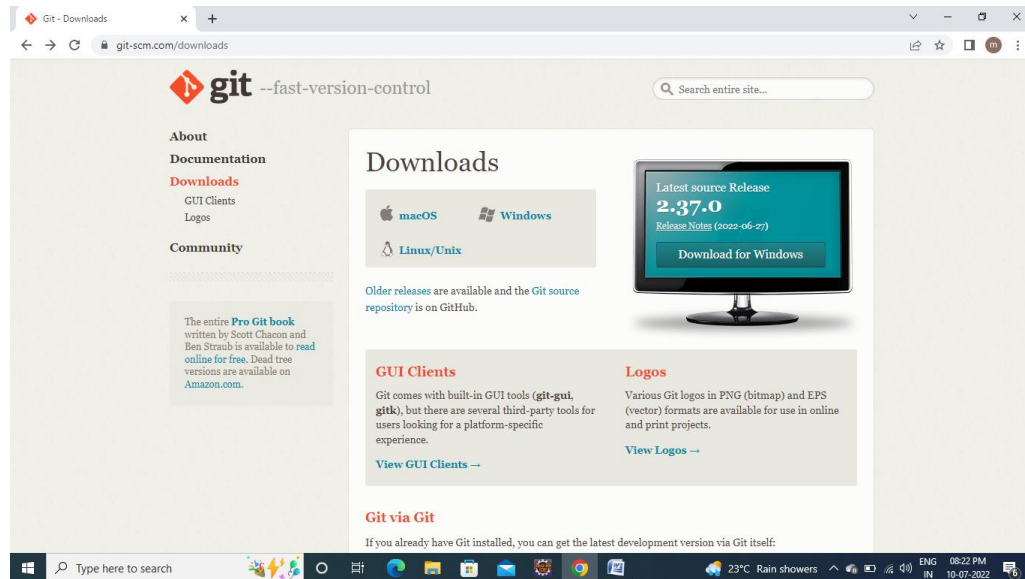
Follow the steps given below to install Git Bash on Windows:

Step 1: Type Git download in browser and press enter

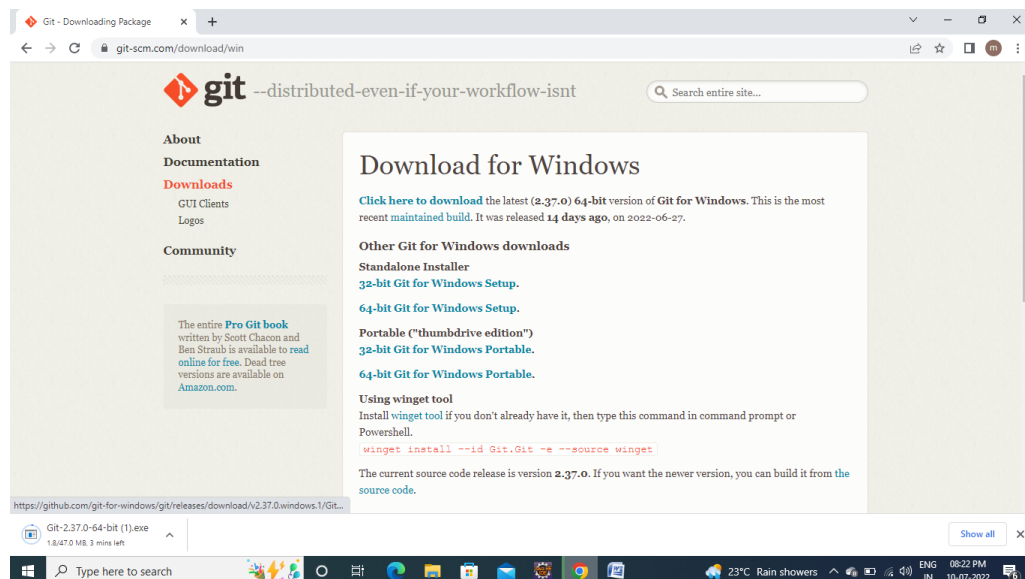
Click on the first link that is “Downloads- Git”.



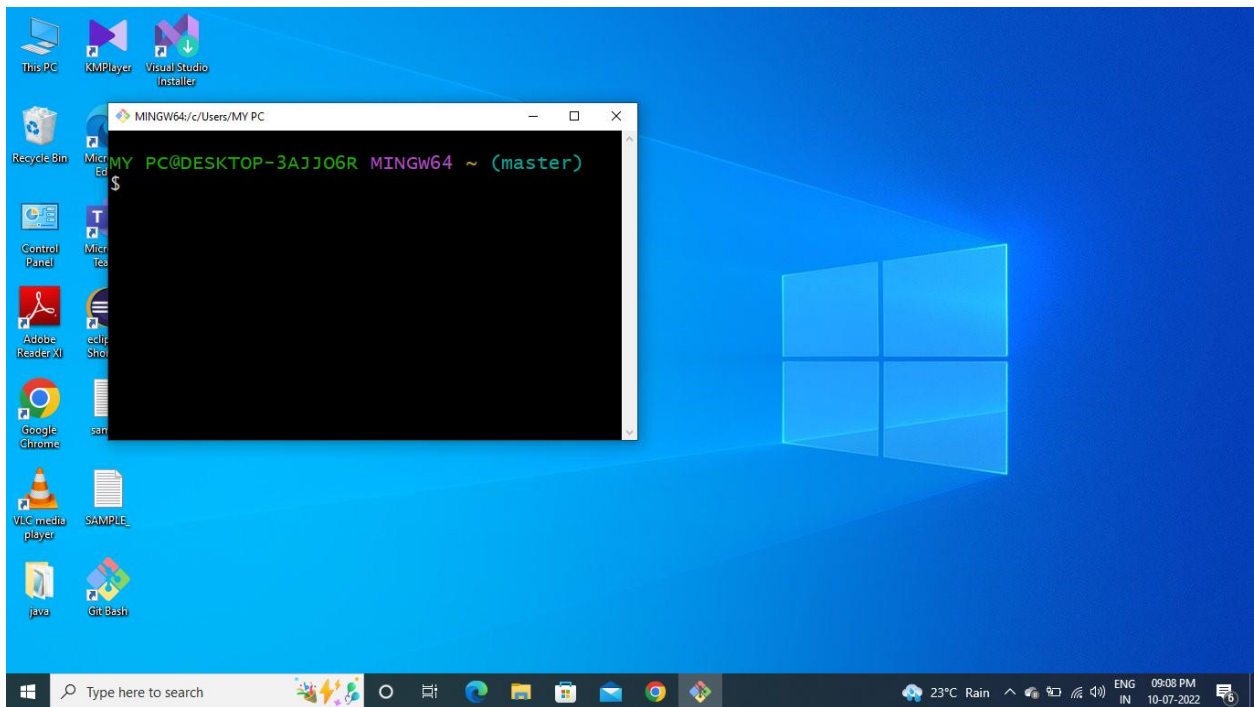
Step 2: Here we are having 3 options macOS, Windows & Linux/Unix. According to your operating system click on the option. In macOS it is already installed.



Step 3: Click on “Click here to download” and it will start downloading.



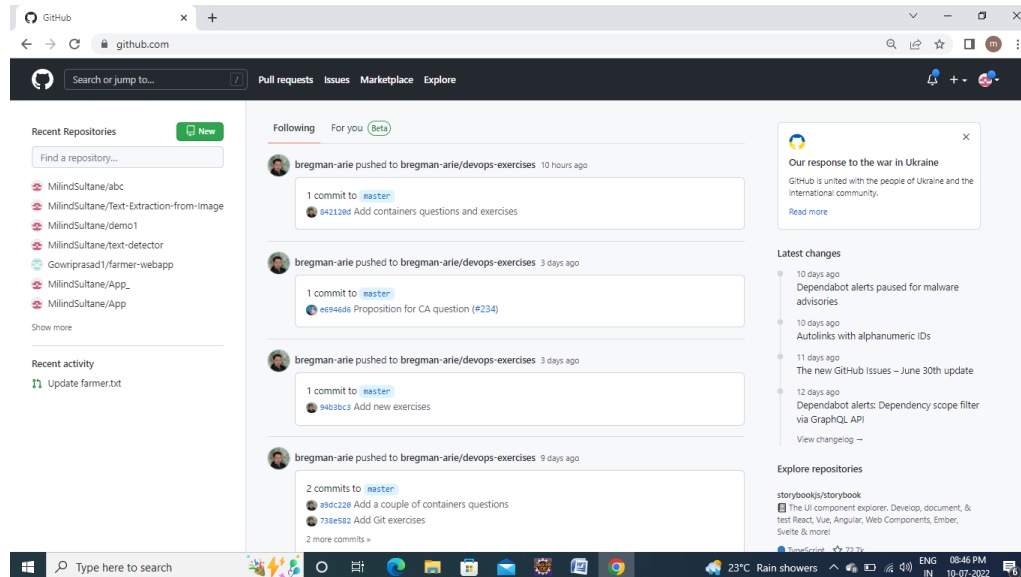
Step 4: Once the downloaded find the included .exe file and open to execute Git Bash.



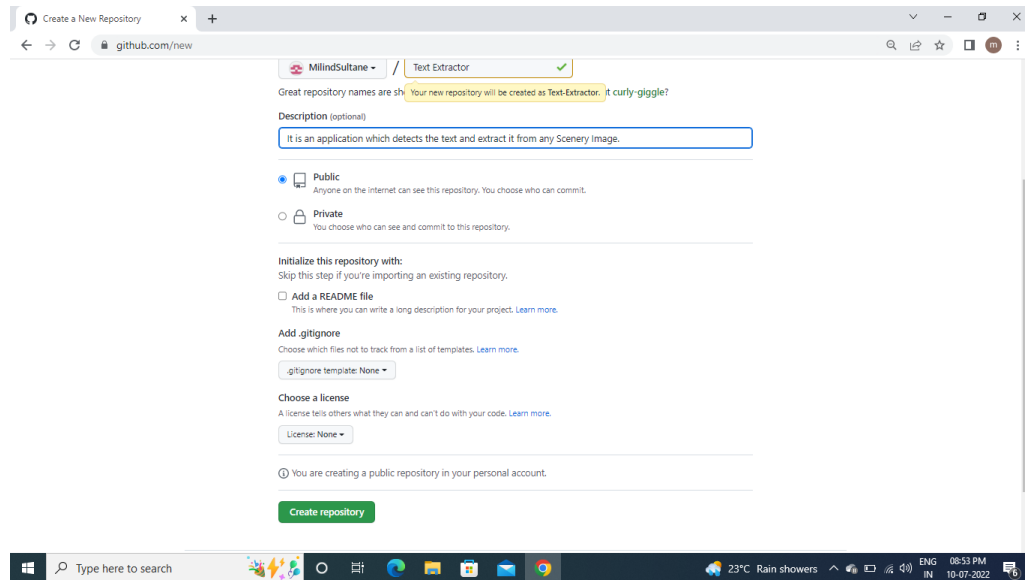
4. Creating & uploading a project on GitHub using Git Terminal

Follow the steps given below to initialize your Local Files/ Repository with Git:

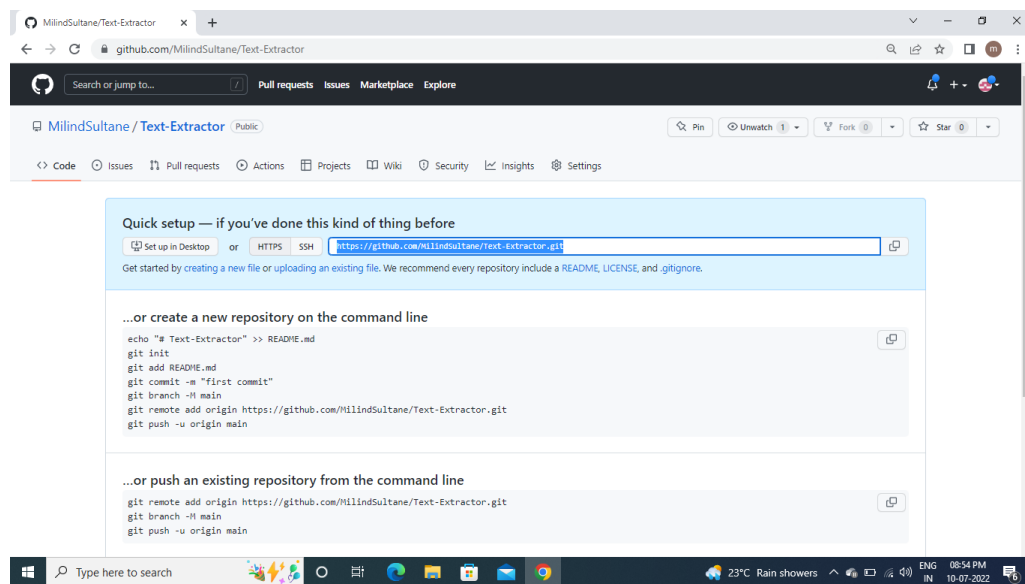
Step 1: Click on the plus icon available on the top right corner. Make a repository on Github.



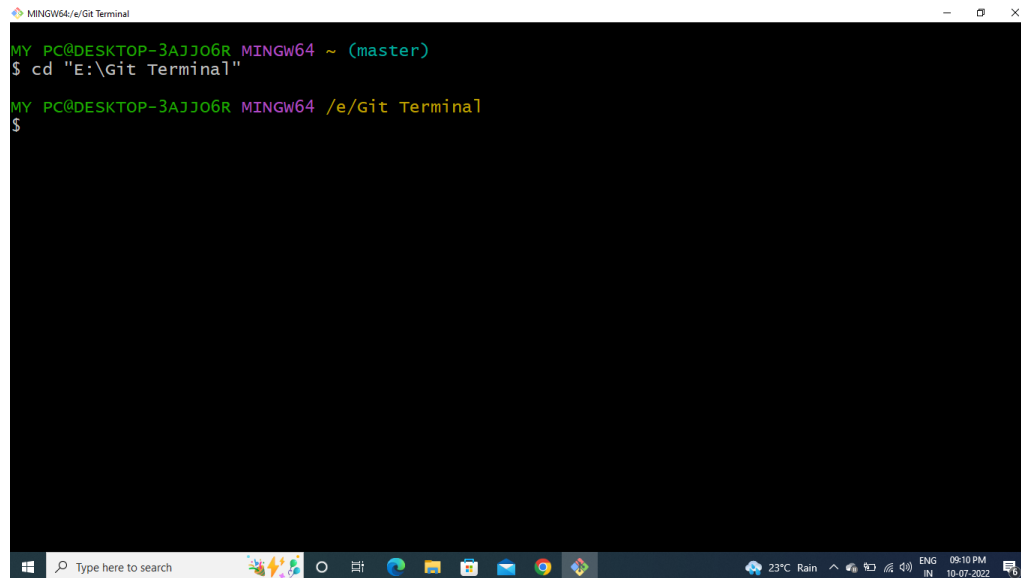
Step 2: Give a suitable name to your repository and click on create repository.



Step 3: The following will appear after creating the repository.



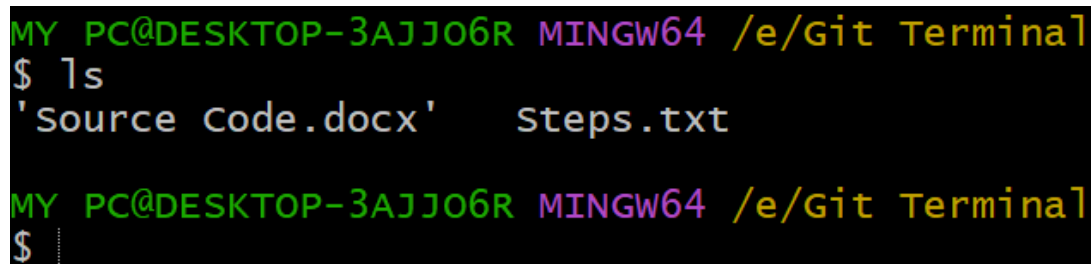
Step 4: Open Git Bash and change the current working directory to your local project by use of **cd** command. Type **cd “File Location”** & press enter.



```
MY PC@DESKTOP-3AJJO6R MINGW64 ~ (master)
$ cd "E:\Git Terminal"

MY PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal
$
```

Step 5: Use **ls** command to check whether the file is added or not. Enter **ls** and press enter. You can see here files from that path are added here.



```
MY PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal
$ ls
'source Code.docx'  steps.txt

MY PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal
$
```

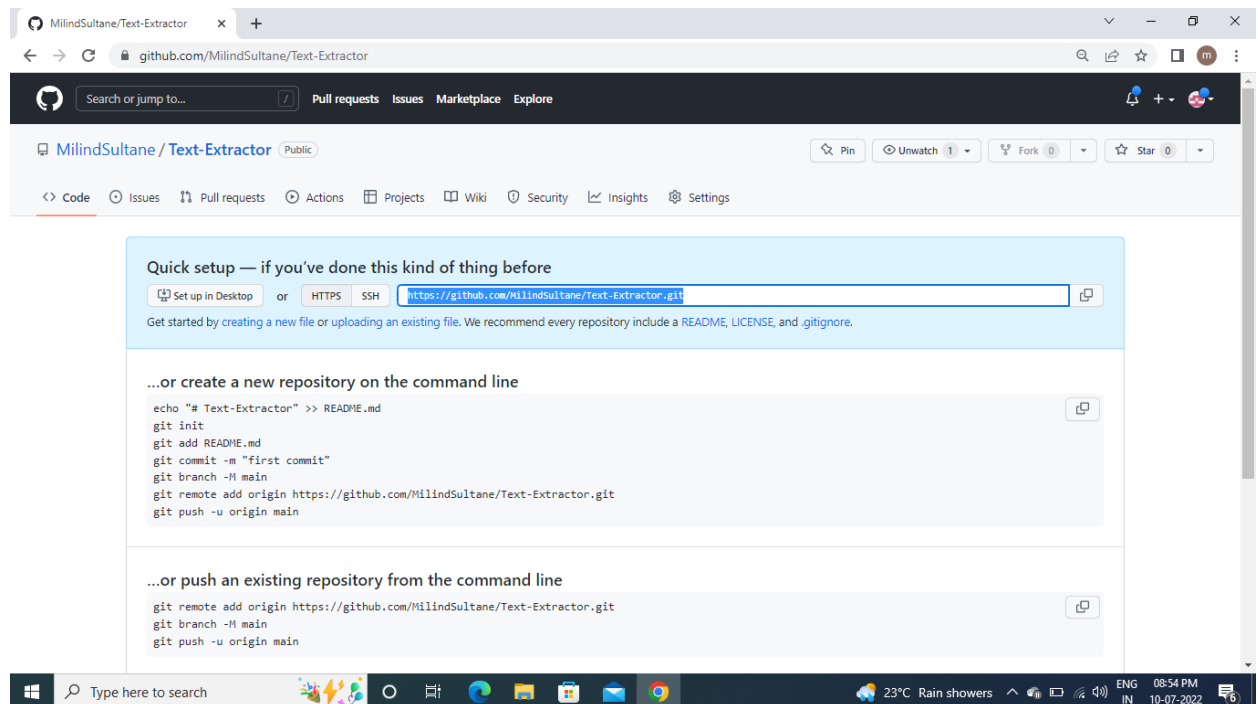
Step 6: Initialize the local directory as a Git repository using **git init** command. Enter **git init** and press enter. It will show the message Initialized empty Git repository in E:/Git Terminal/.git/

Now the GitHub is initialized.

```
MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal
$ git init
Initialized empty Git repository in E:/Git Terminal/.git/

MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ |
```

Step 8: Now go to Github repository and in code section copy the HTTPS URL. Then go to the git bash again.



Step 9: To push our source code or files to our repository use git remote add origin command. Type `$ git remote add origin “paste https URI”` and press enter.

```
MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ git remote add origin https://github.com/MilindSultane/Text-Extractor.git

MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ git remote add origin https://github.com/MilindSultane/Text-Extractor.git

MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ |
```

Step 10: To check whether we have access to push and pull. Use command `git remote -v`. Type `git remote -v` and then press enter. Now we can see here we are able to fetch and push.

```
MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ git remote -v
origin https://github.com/MilindSultane/Text-Extractor.git (fetch)
origin https://github.com/MilindSultane/Text-Extractor.git (push)

MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ |
```

Step 11: Use `git add .` command to add our files. Type `git add .` and press enter.

```
MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ git add .

MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ |
```

Step 12: Commit the files that you've staged in your local repository using **git commit -m**. Type **git commit -m "First Commit"** and click enter.

```
MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ git commit -m "First Commit"
[master (root-commit) 56a0285] First Commit
 2 files changed, 5 insertions(+)
 create mode 100644 Source Code.docx
 create mode 100644 Steps.txt

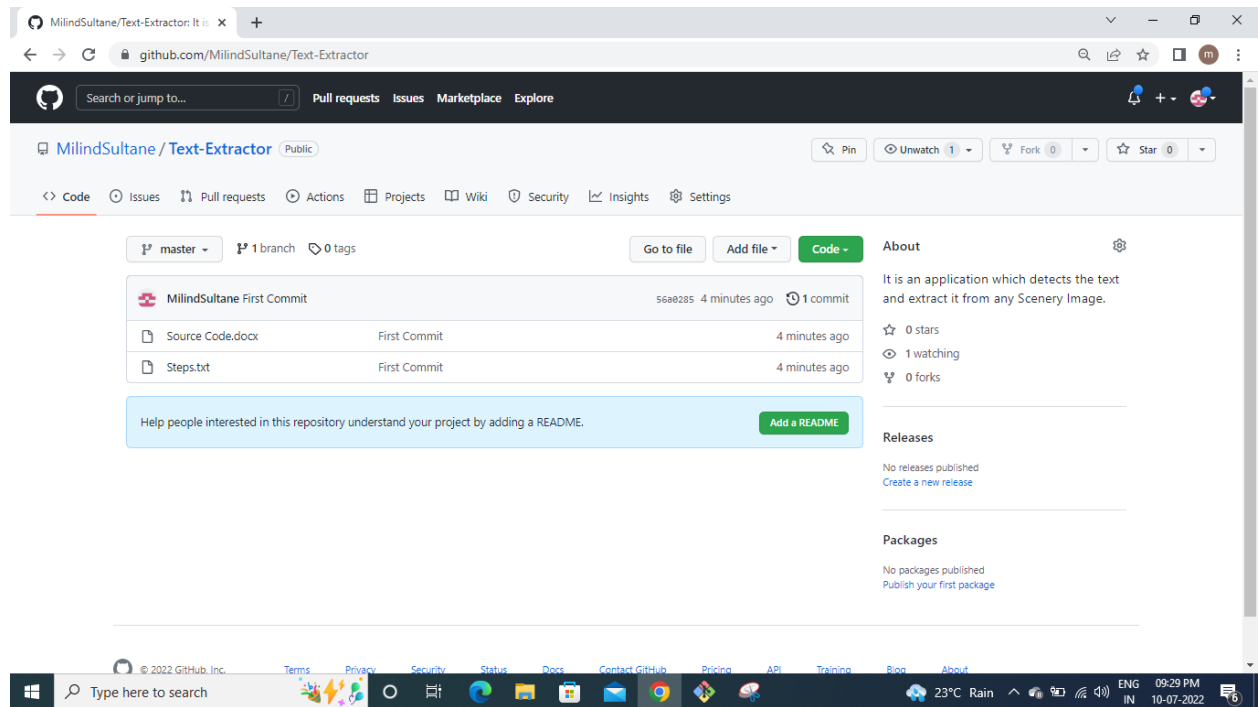
MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ |
```

Step 13: Finally we have to use **git push origin master** command so that it will push our files to the repository. Type **git push origin master** and press enter. When you are doing it first time it will ask you for your GitHub login credential.

```
MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 23.28 KiB | 7.76 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MilindSultane/Text-Extractor.git
 * [new branch]      master -> master

MY_PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
```

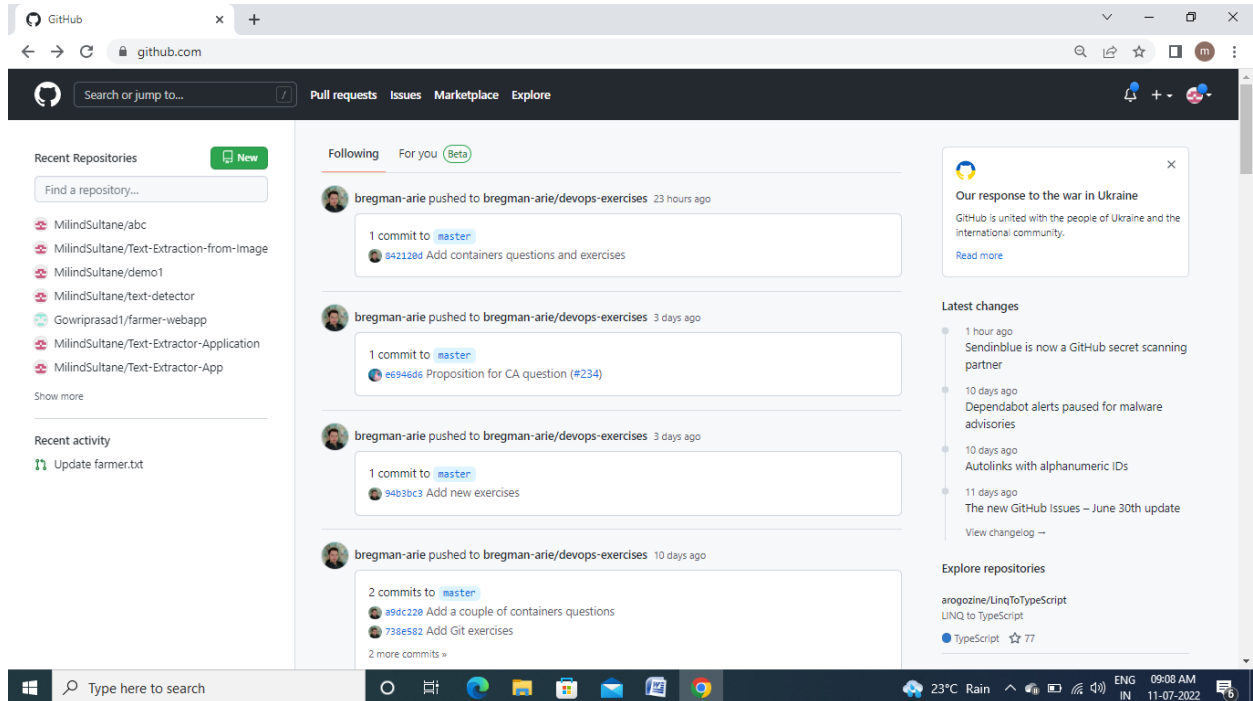
Step 14: Go to GitHub and refresh the page then new changes will updated there.



Now we can see that the files we want to push are updated here successfully.

5. Creating & uploading a project on GitHub using Git GUI

Step 1: Sign in in GitHub account. Click on the plus icon available on the top right corner. Make a repository on Github.



Step 2: Give a suitable name to your repository and click on create repository.

Create a New Repository

Owner ^{*} MilindSultane / Repository name ^{*} Text_Extractor ✓

Great repository names are short, descriptive, and unique. Text_Extractor is available. [?] inspiration? How about furry-parakeet?

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:
Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None

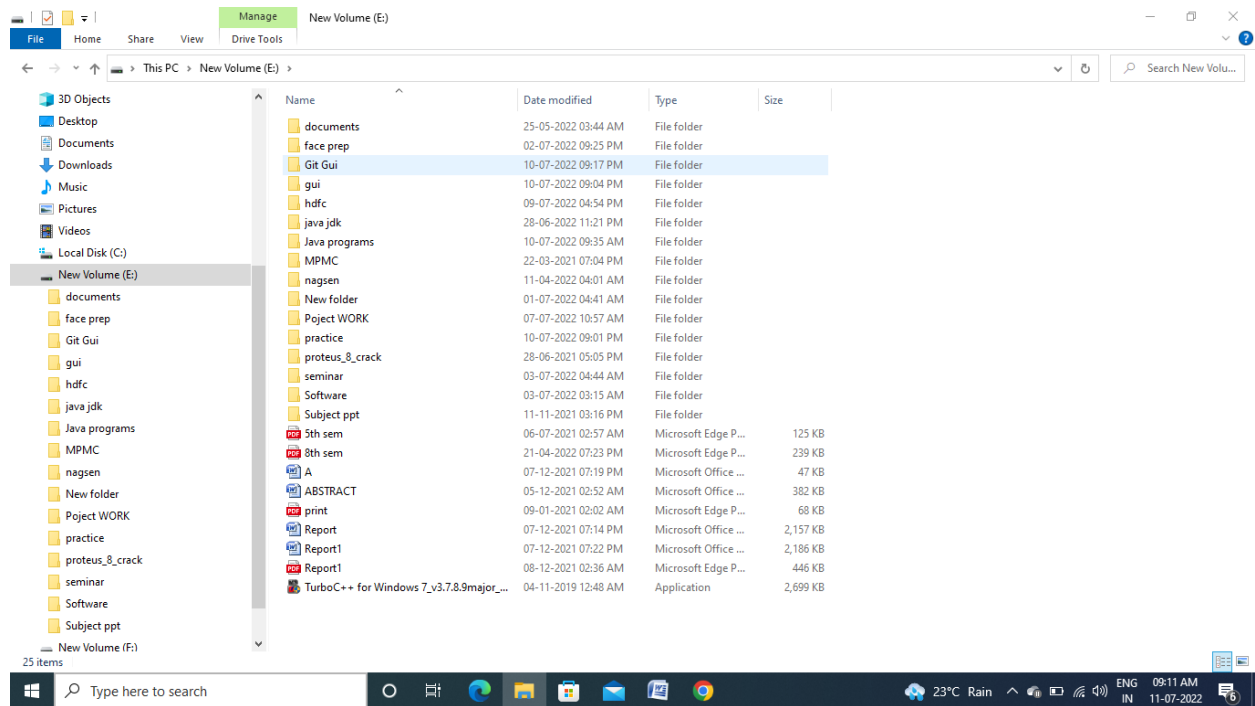
Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None

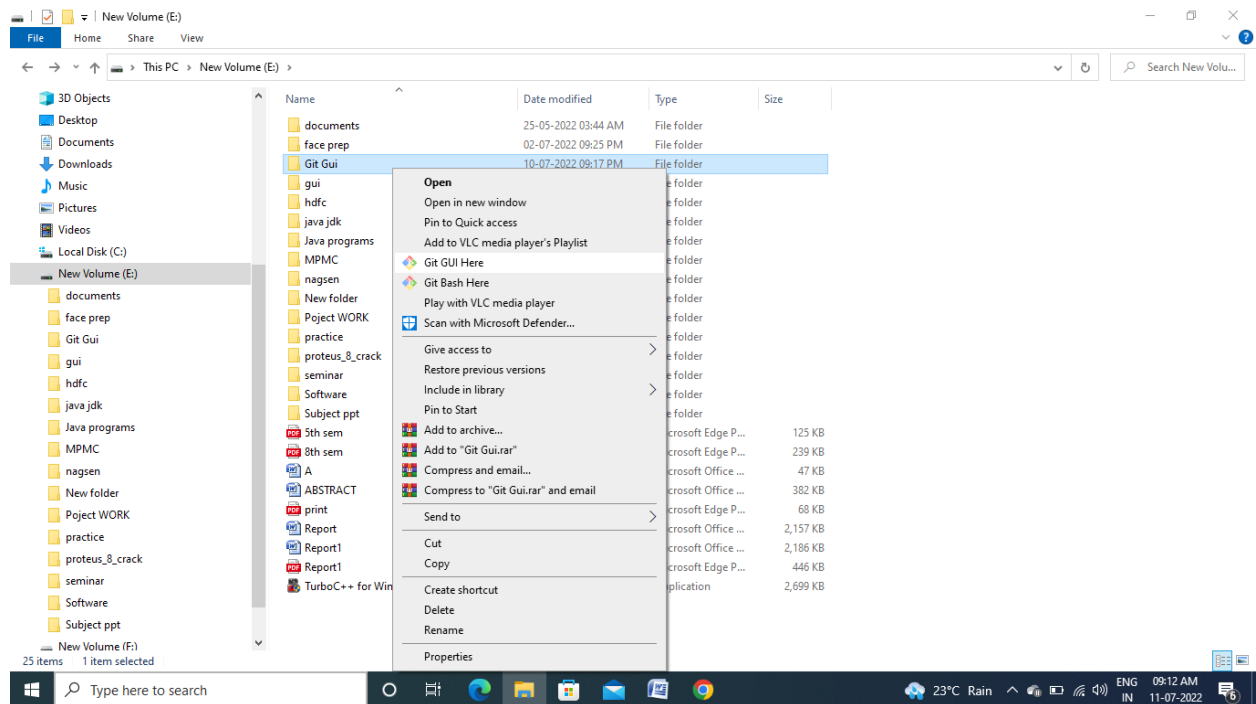
☐ You are creating a public repository in your personal account.

Create repository

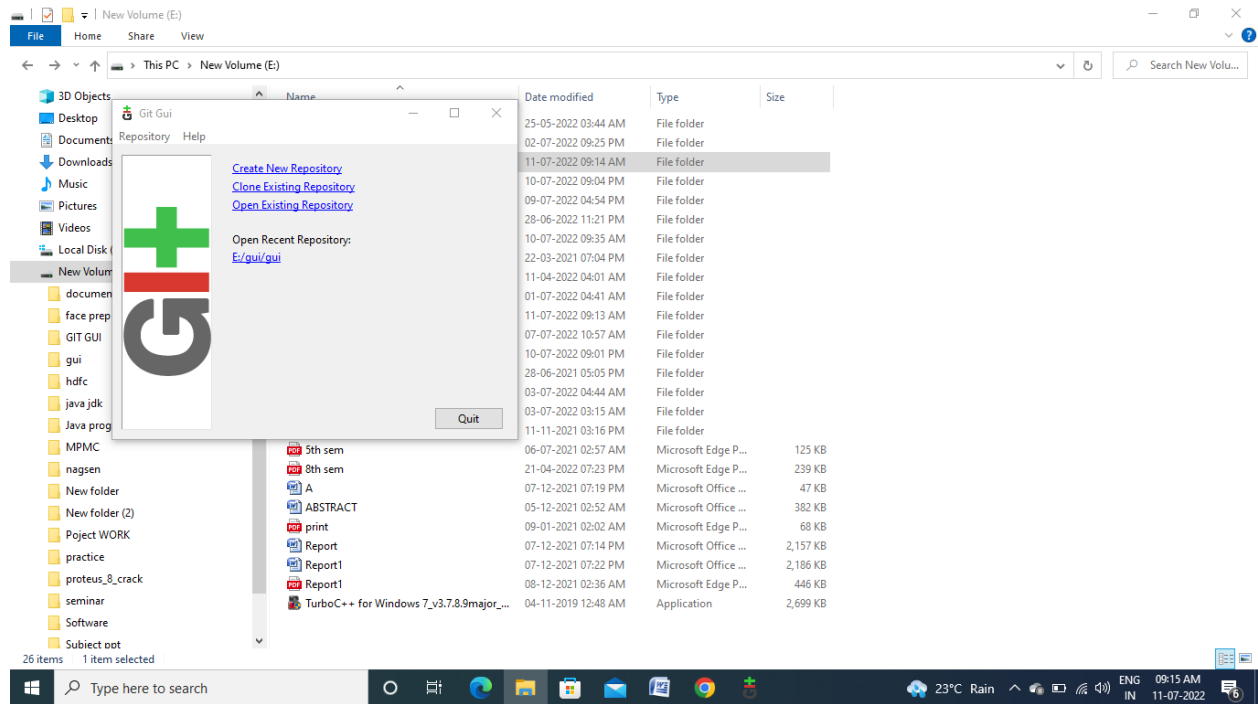
Step 3: Make any folder at any drive in your system and then right click.



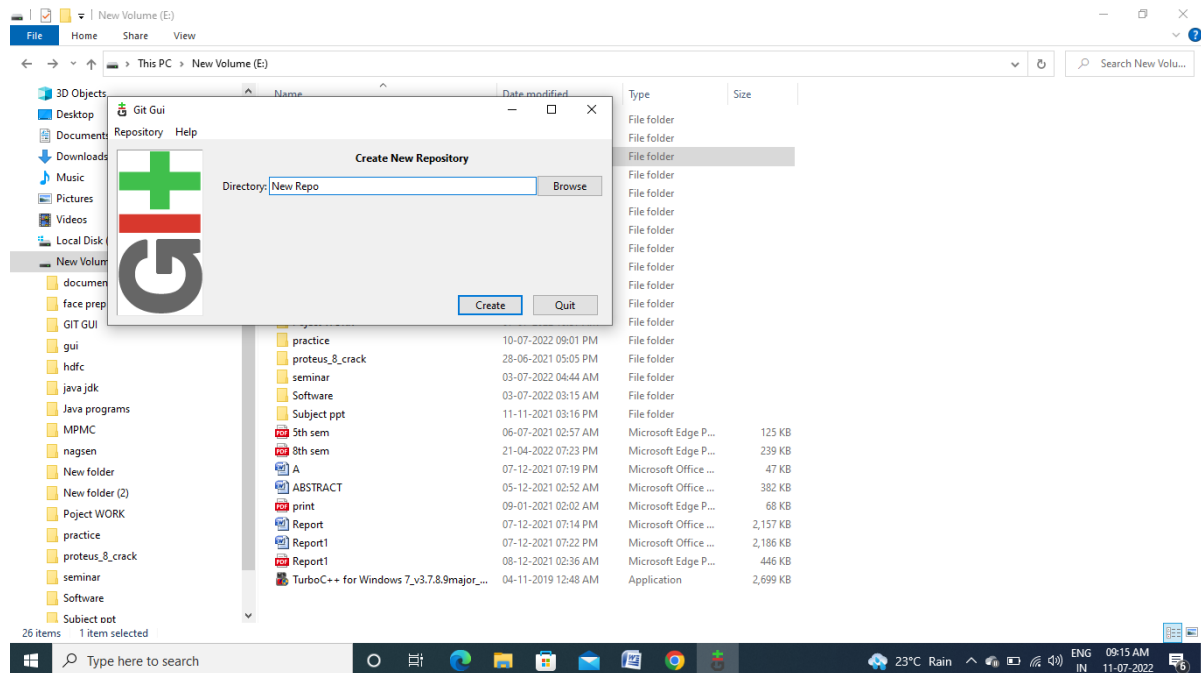
Step 4: Right click on created folder and click Git GUI here.



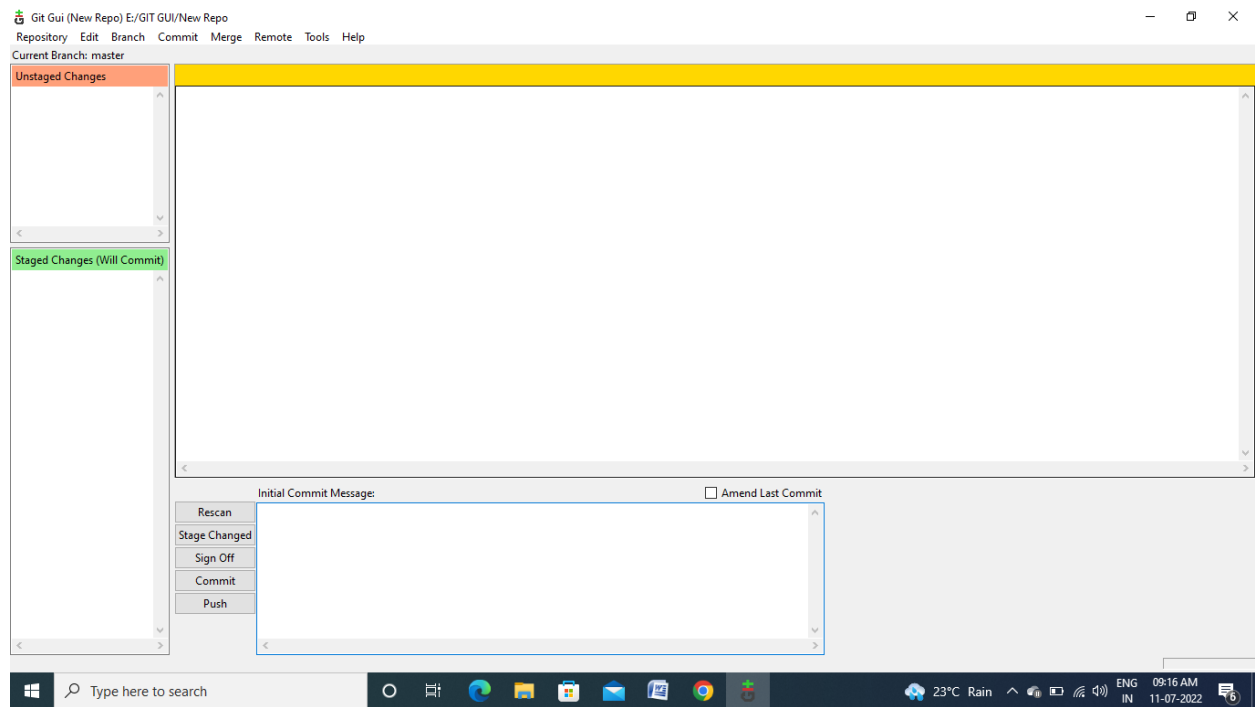
Step 5: Click on Create New Repository.



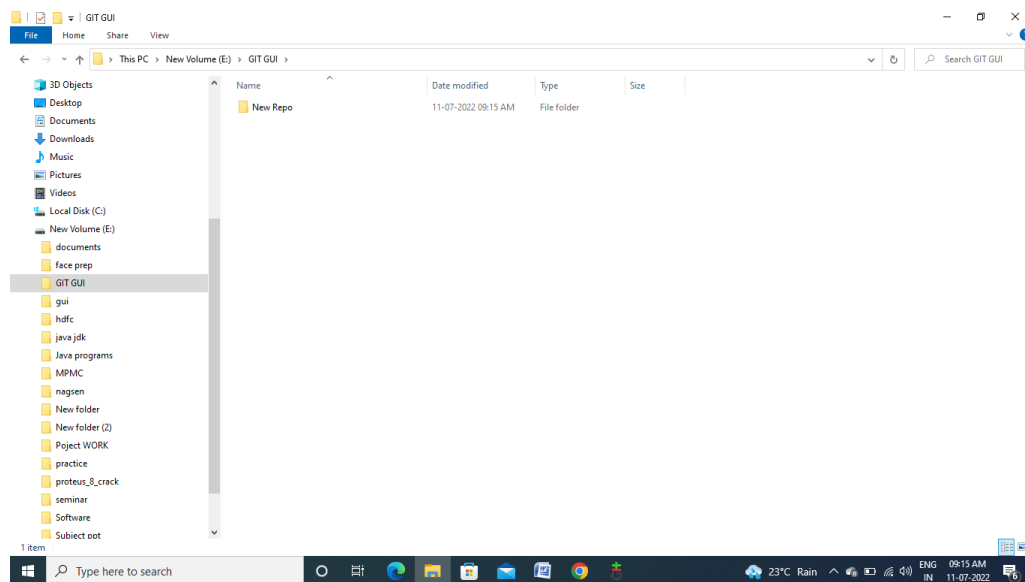
Step 6: Give the suitable name to the Directory and click on Create.



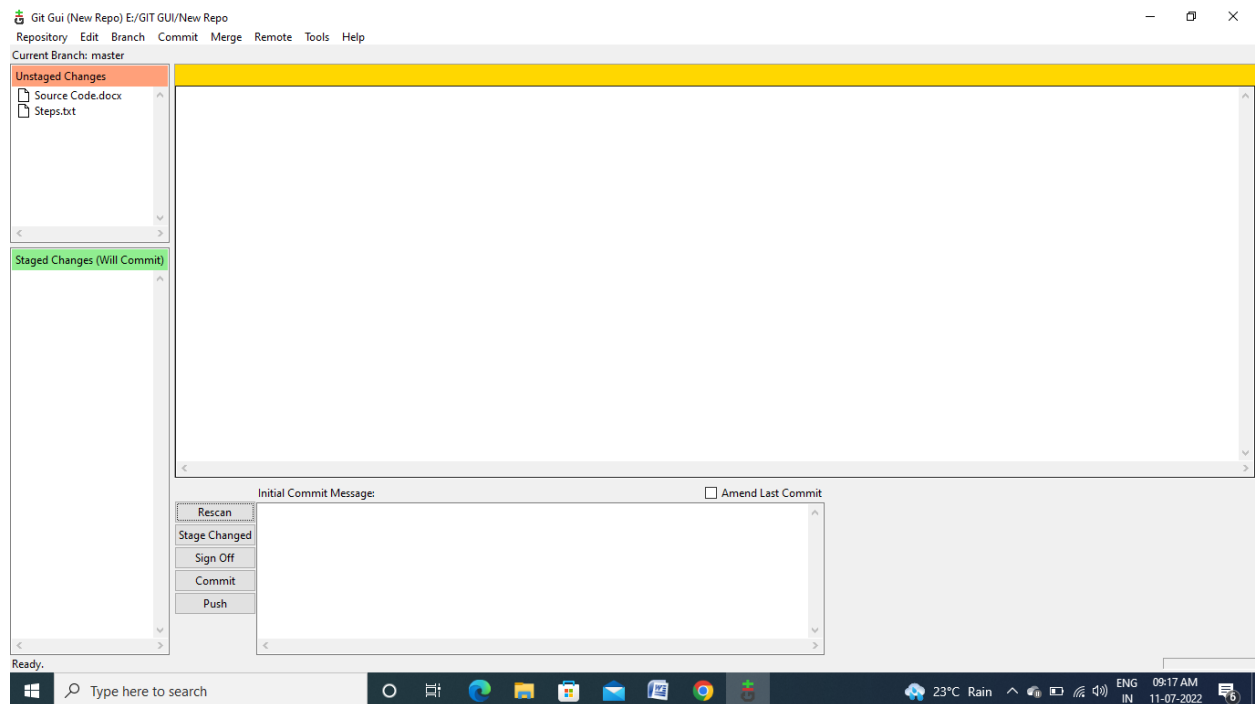
Step 7: After creating the directory this interface will open.



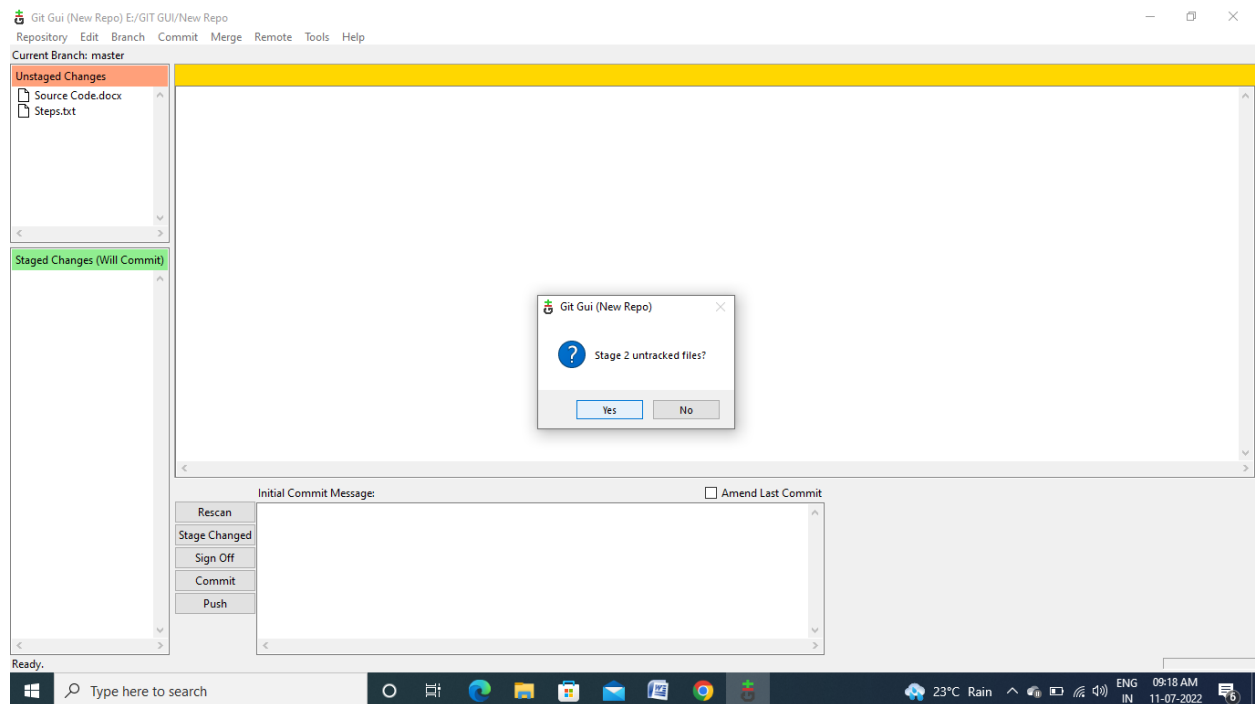
Step 8: Now the new directory is created. Now add your project/work/ files to this folder.



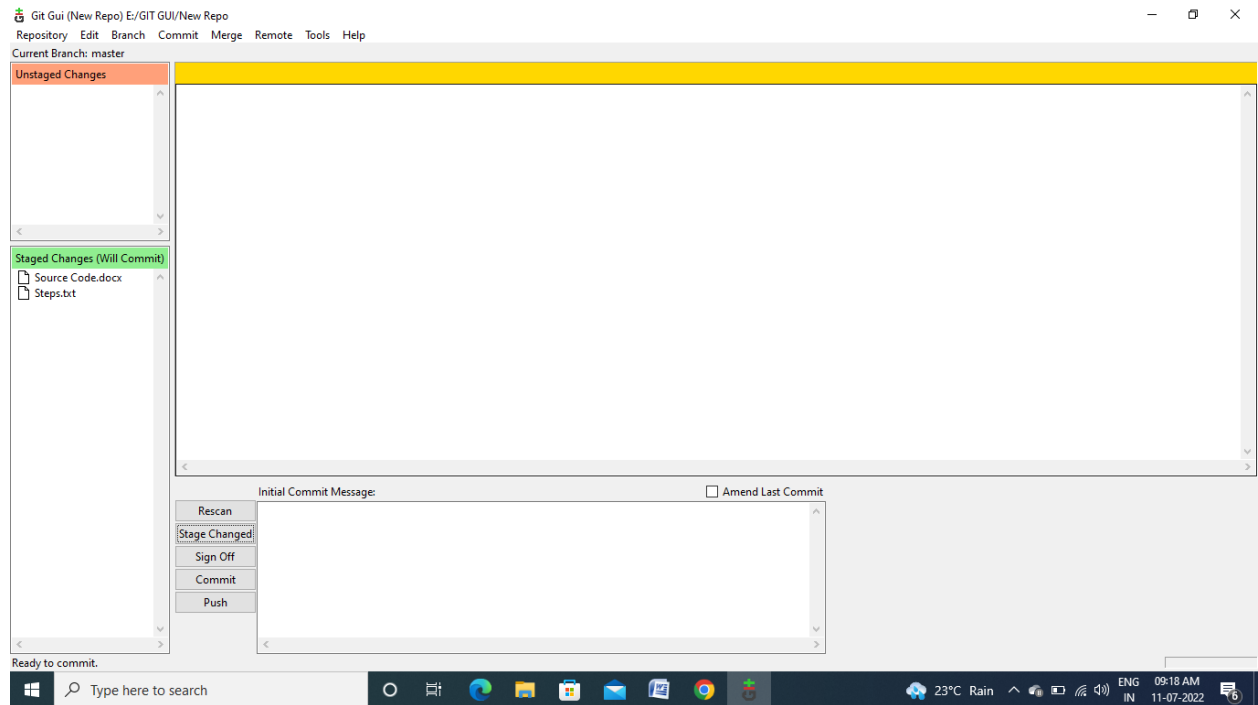
Step 9: Go to Git GUI and click on Rescan button. Now the files are seen in Unstaged Changes section.



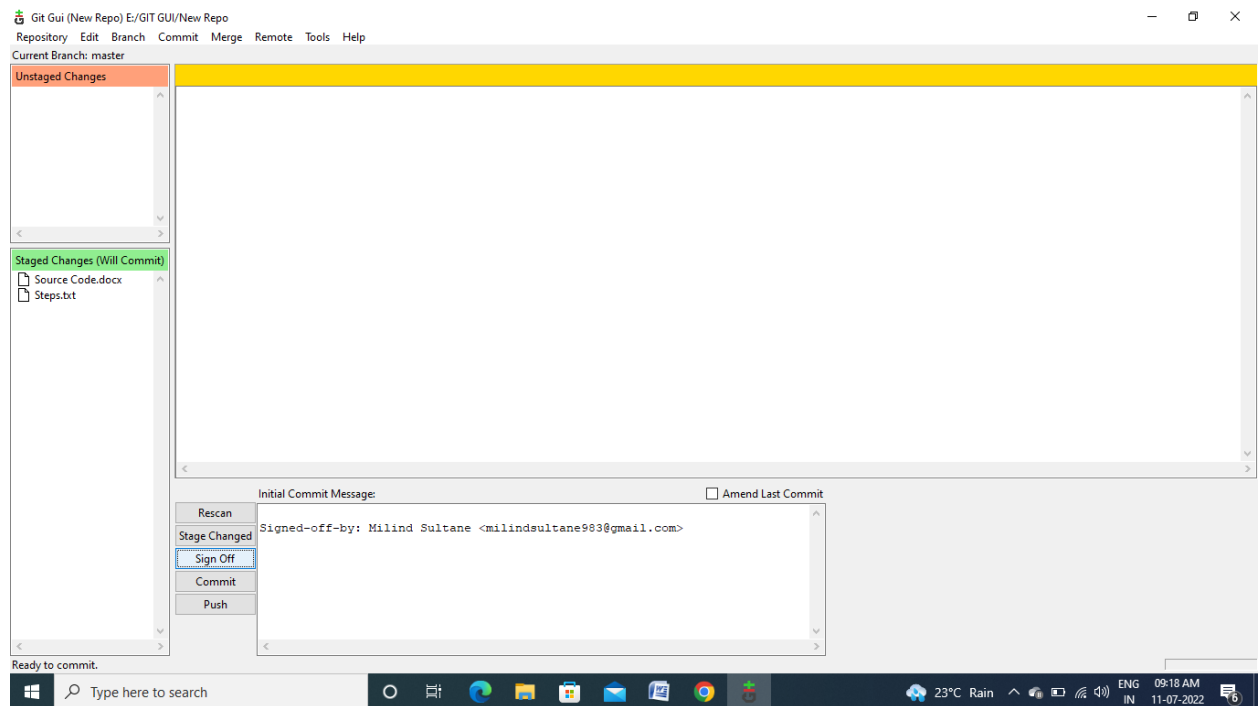
Step 10: To push the files we have to move our files from Unstaged Changes to the Staged Changes section. Click on Stage Changed button & click Ok to the pop up.



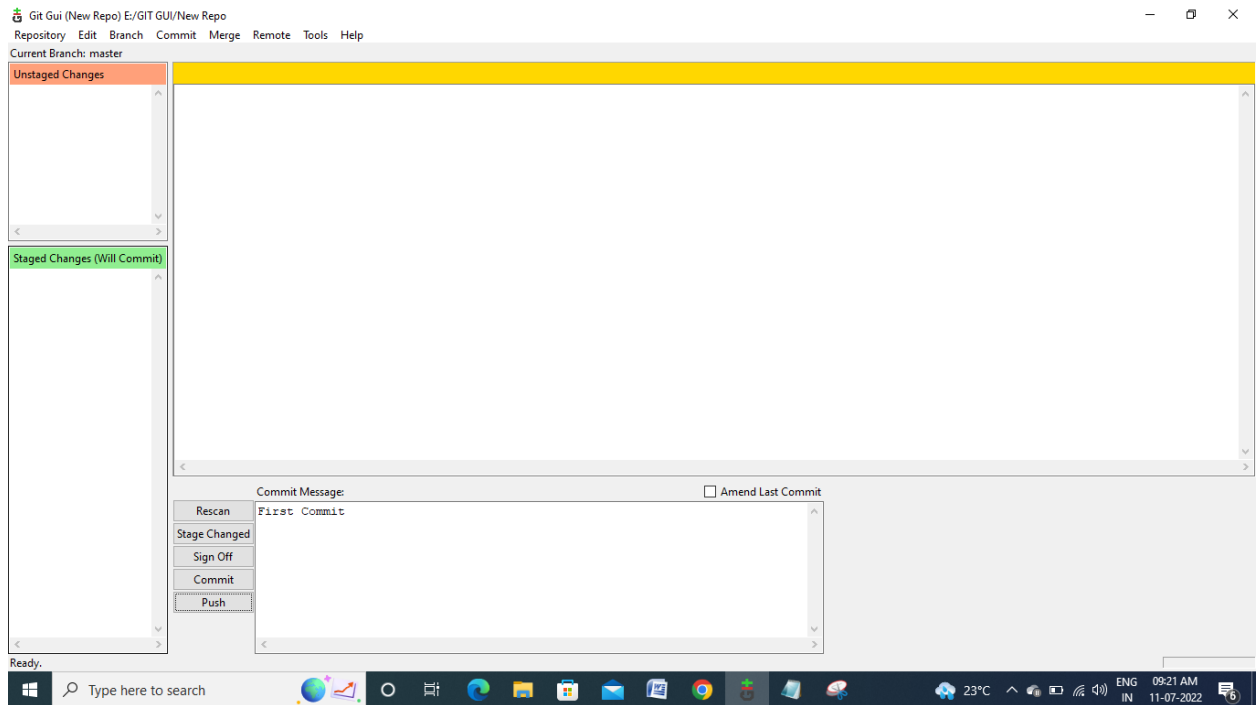
Step 11: Then we can see here files moved from Unstaged Changes to Staged Changes



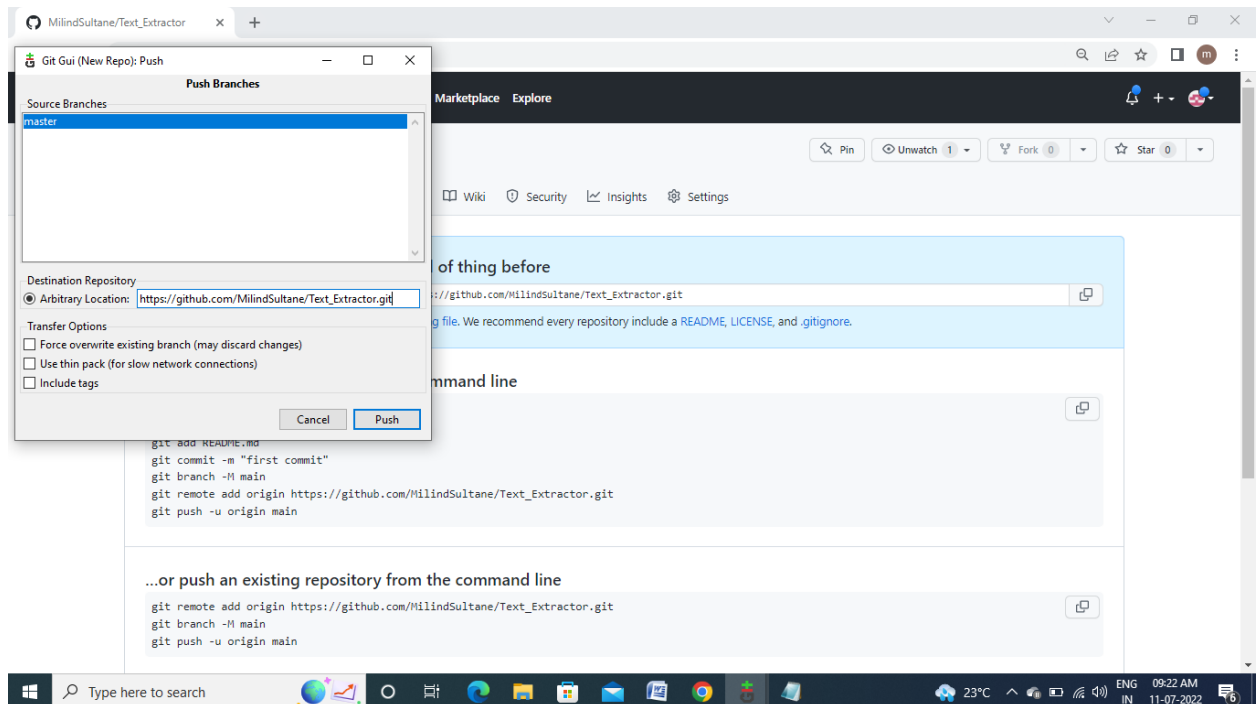
Step 12: Sign off shows the information about the user who commit the changes. It will shows the name and email address of the user.



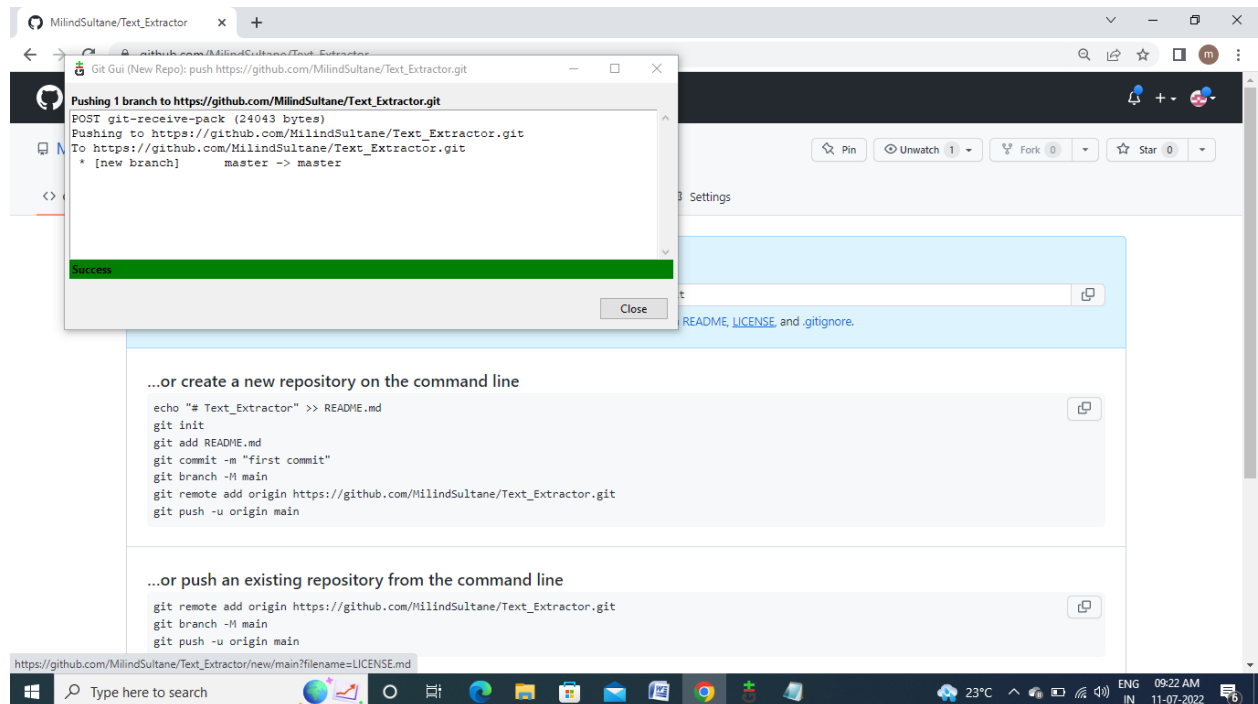
Step 13: Commit whatever you want to commit and then click on Commit button.



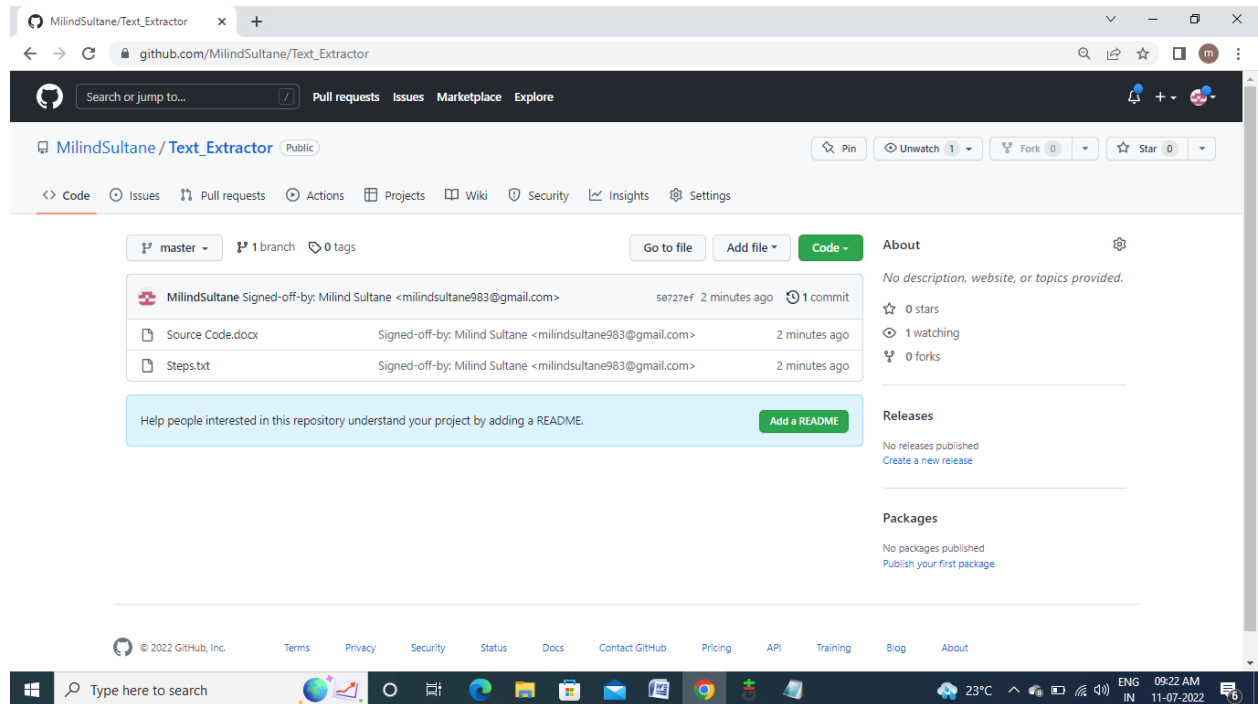
Step 14: Click on Push button. After clicking on Push button it will ask you arbitrary location. Then go to the GitHub copy the https link and paste here.



Step 15: First time it will ask you for login credential. Enter your username and password. Then it will show success. Then click on close



Step 16: Go to GitHub and refresh the page then new changes will updated there.



6. Conclusion

In this documentation we have successfully created and uploaded our project/ files from our local system on GitHub using both Git Terminal and Git GUI. Both has been successfully completed.