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 make separate changes to web pages at the same time. As Carpenter notes, because 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

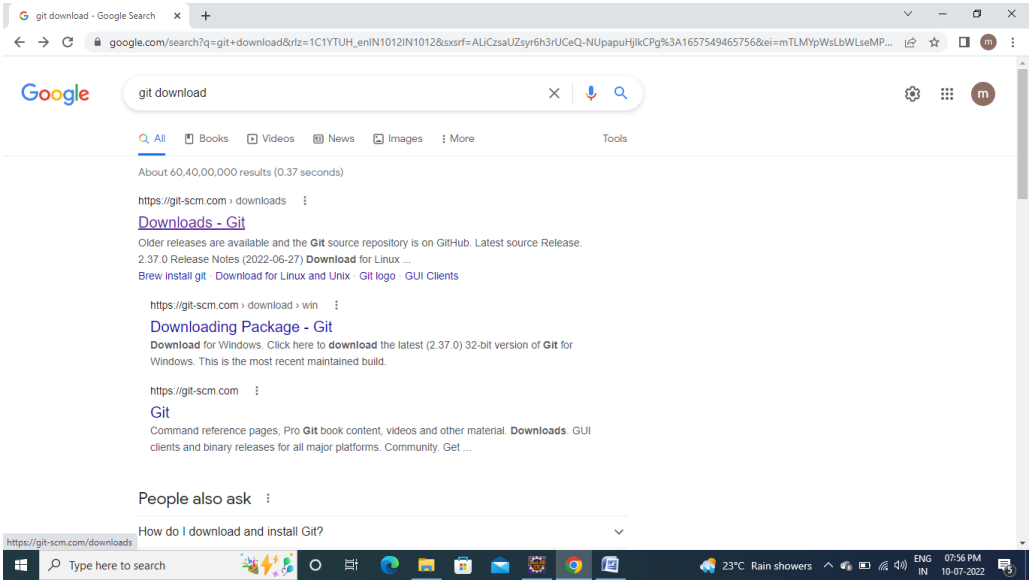
Git Bash is an application that provides Git command line experience on the Operating System. It is a command-line shell for enabling git with the command line in the system. A shell is a terminal application used to interface with an operating system through written commands. Git Bash is a package that 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:

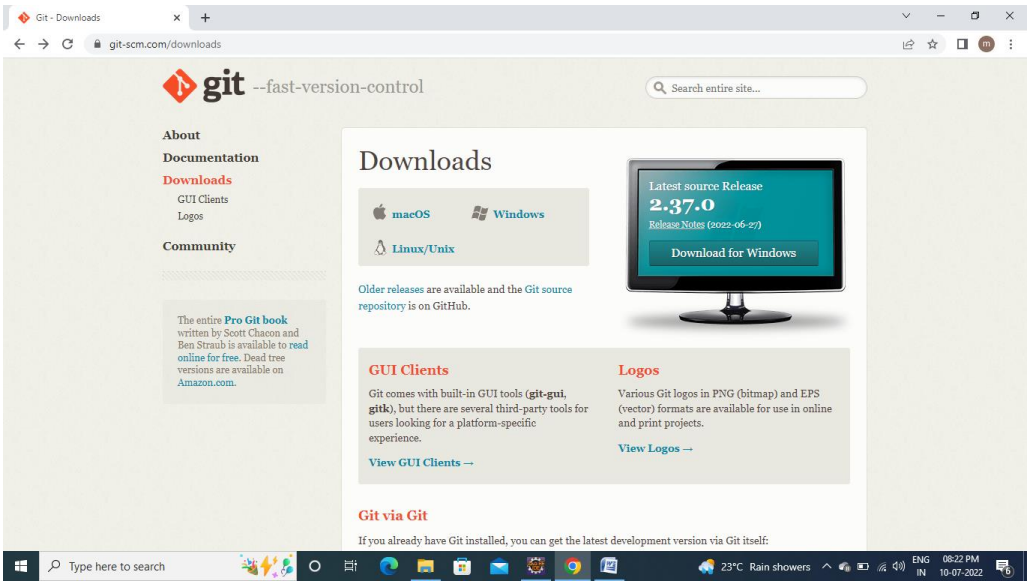
Step 1: Type Git download in browser and press enter

Click on the first link that is "Downloads- Git".

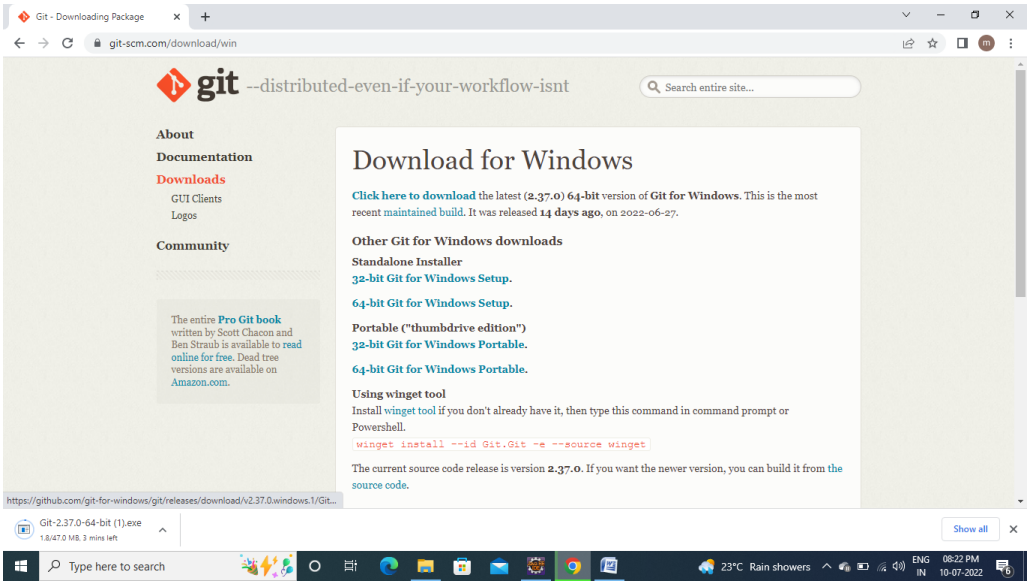


Step 2: Here we are having 3 options macOS, Windows & Linux/Unix

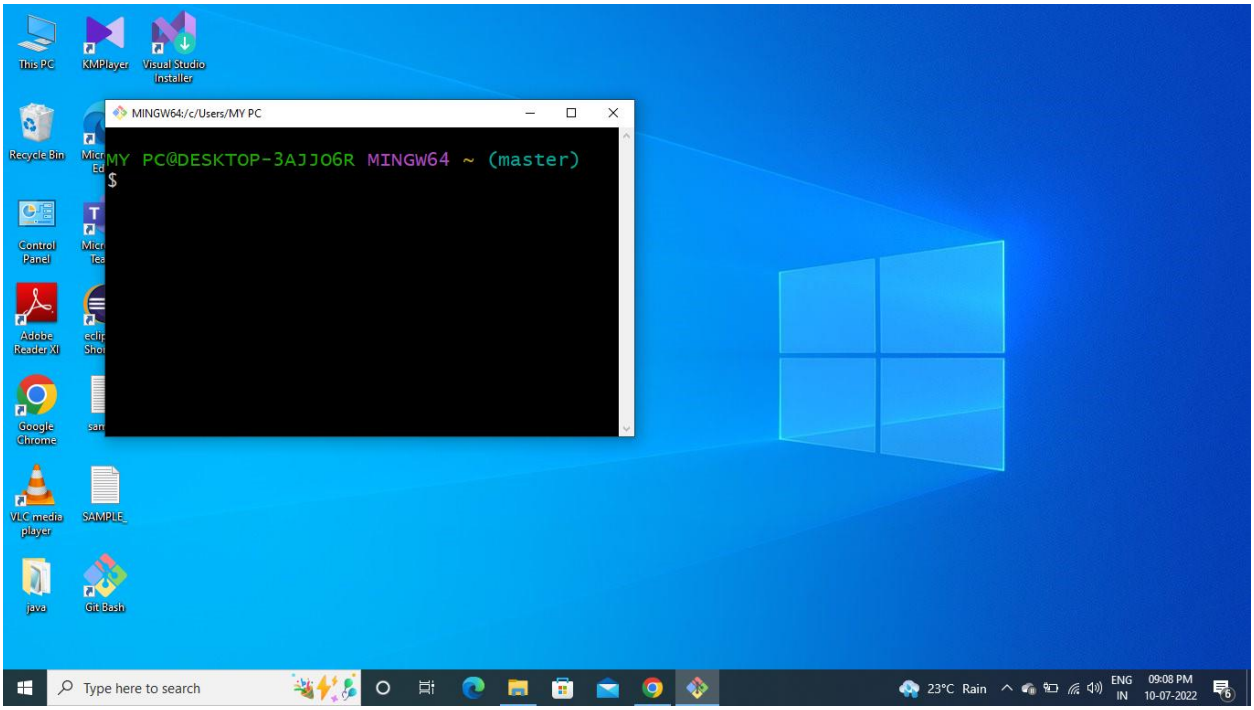
According to your operating system click on the option. In macOS it is already installed.



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.

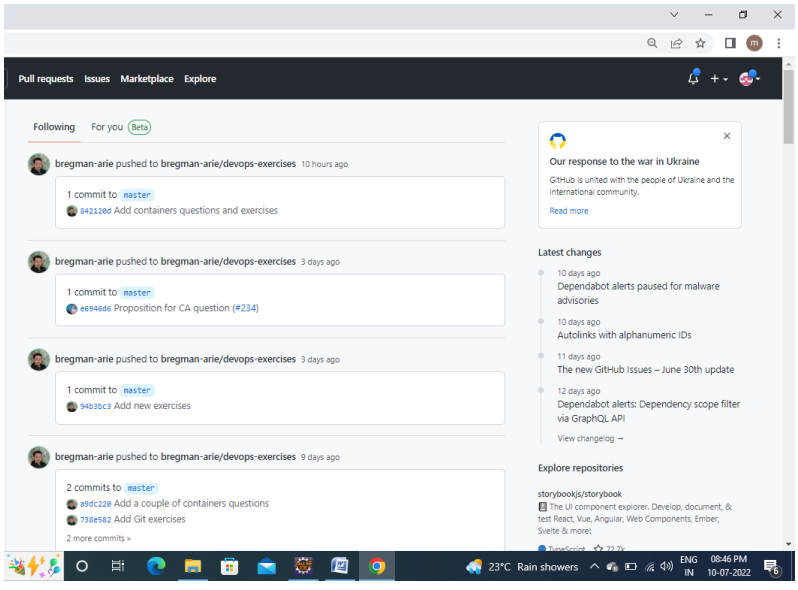


4. Initializing a Local repository using Git Terminal

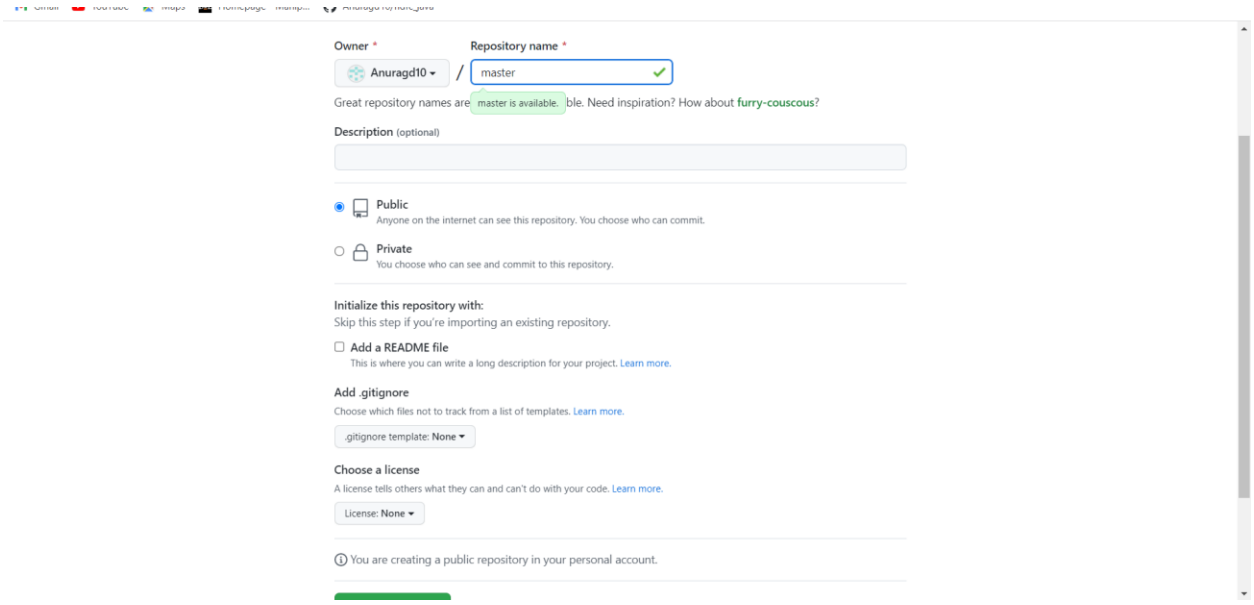
.

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

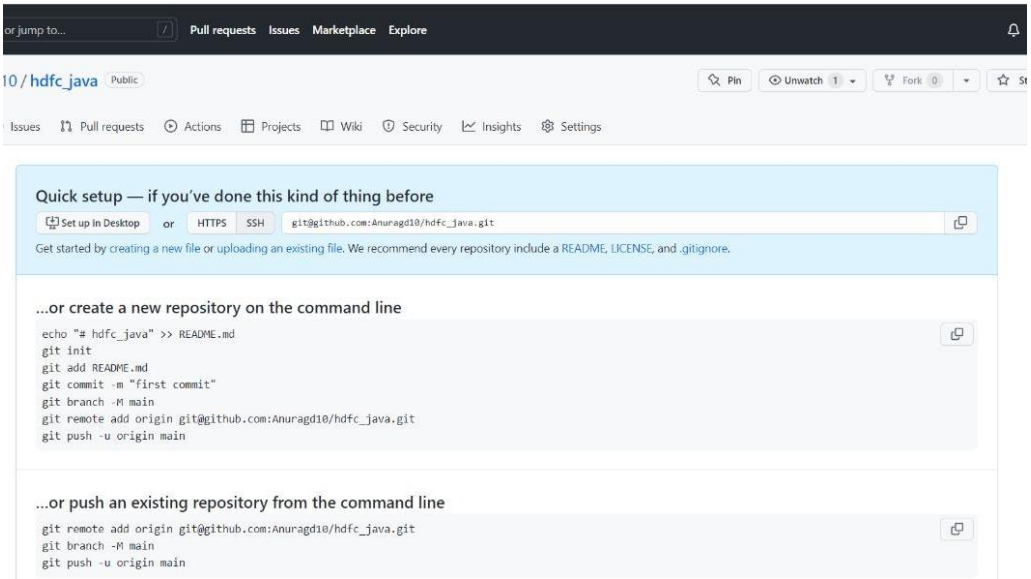
Step 1: Click on the plus icon available on the top right corner. Make a repository on Github.



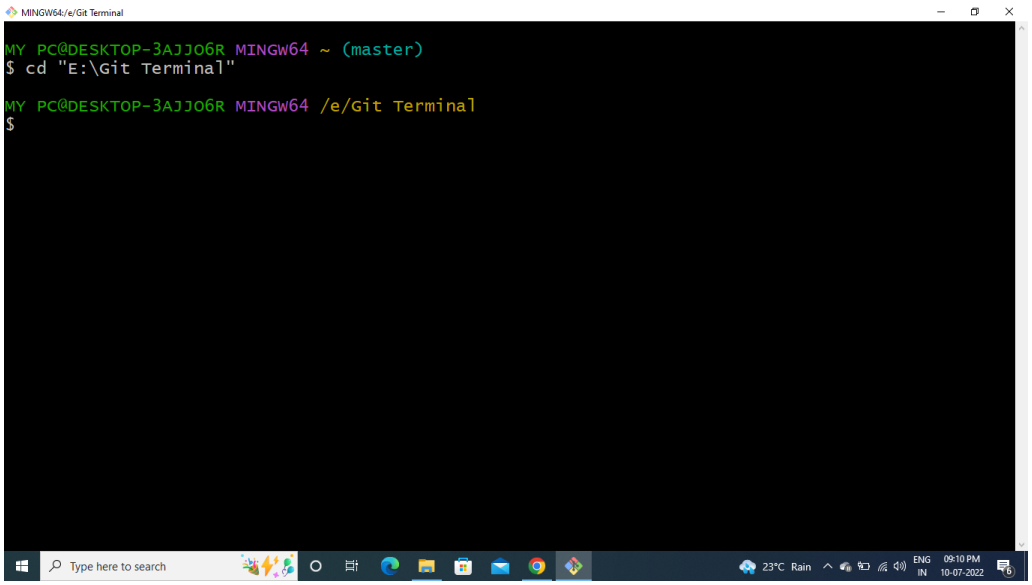
Step 2: Give a suitable name to your repository and click on create repository.



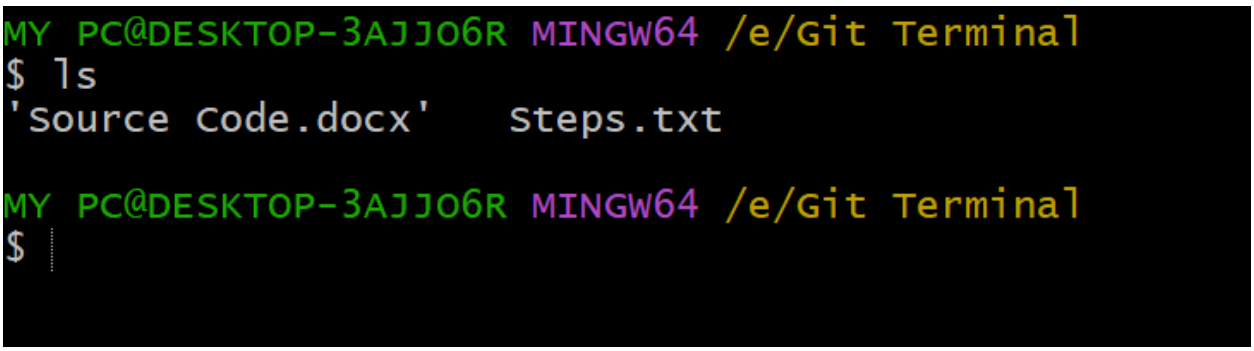
Step 3: The following will appear after creating the repository



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.

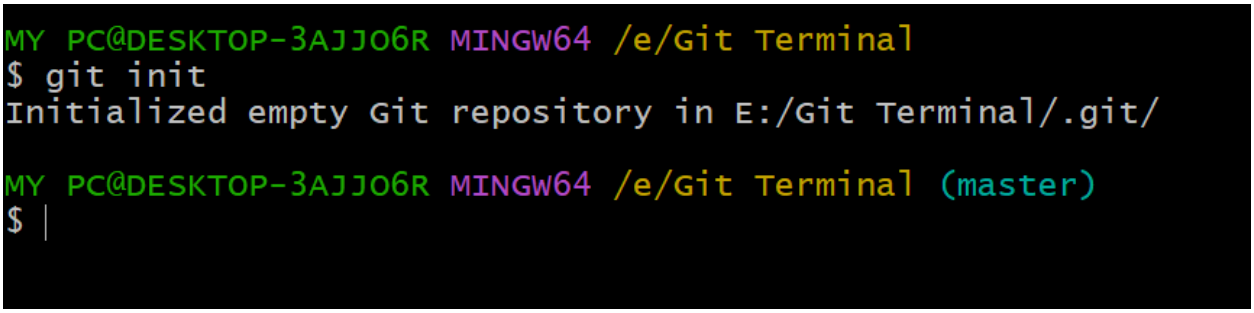


Step 5: Use ls command to check weather the file is added or not. Enter ls and press enter. You can see here files from that path are added here.

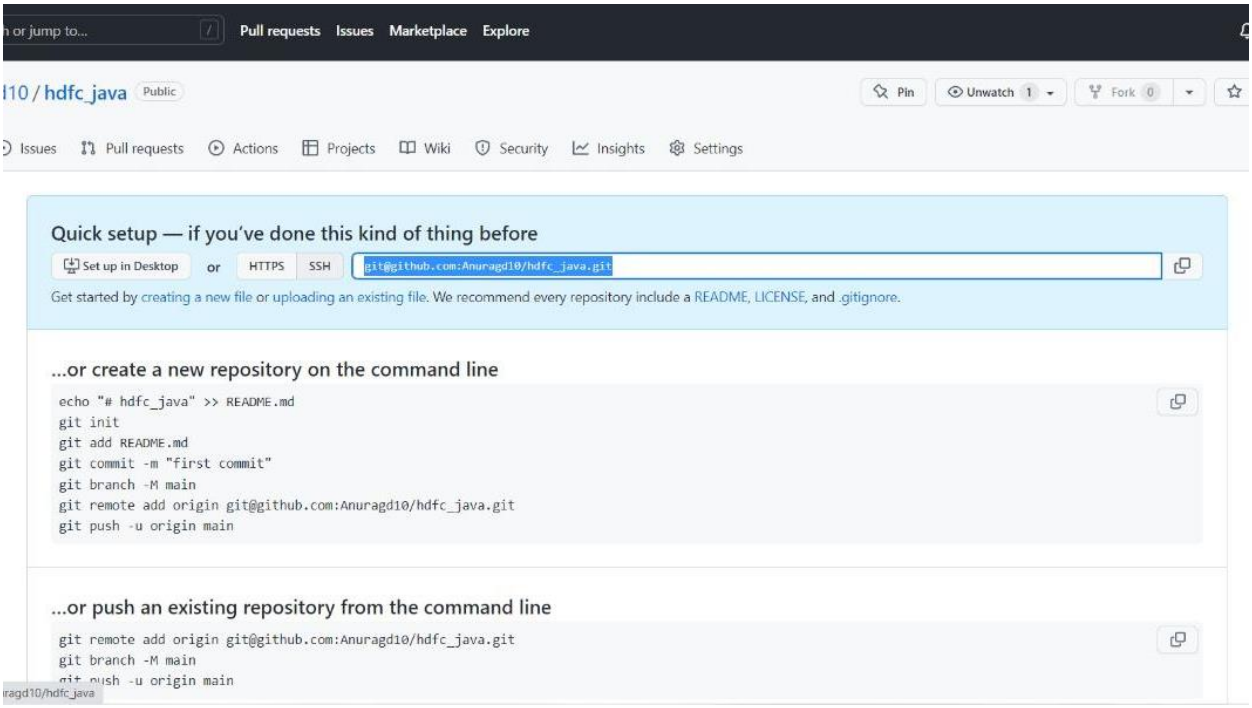


Step 6: Initialize the local directory as a Git repository using git init command. Enter git init and press enter. It will show the message Initialized empty Git repository in E:/Git Terminal/.git/

Now the GitHub is initialized.

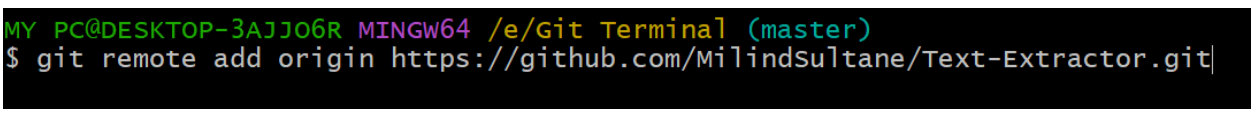


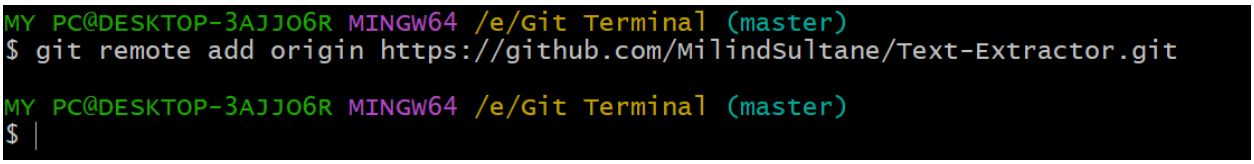
Step 8: Now go to Github repository and in code section copy the HTTPS URL. Then go to the git bash again.



Step 9: To push our source code or files to our repository use git remote add origin command.

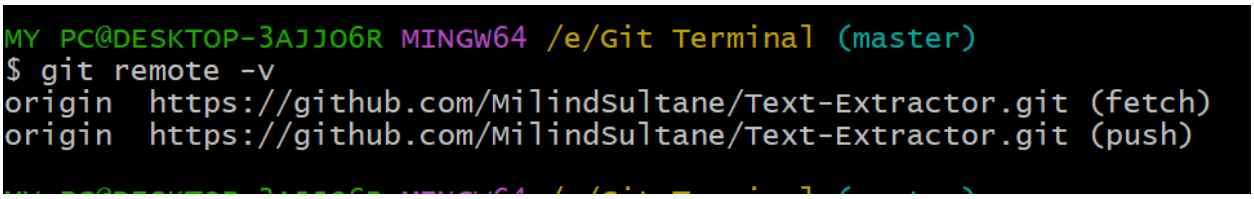
Type $ git remote add origin "paste https URl" and press enter.



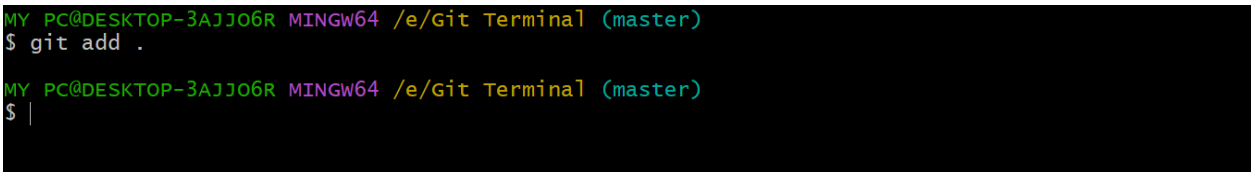


Step 10: To check weather we have access to push and pull. Use command git remote –v.

Type git remote –v and then press enter. Now we can see here we are able to fetch and push.



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



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.

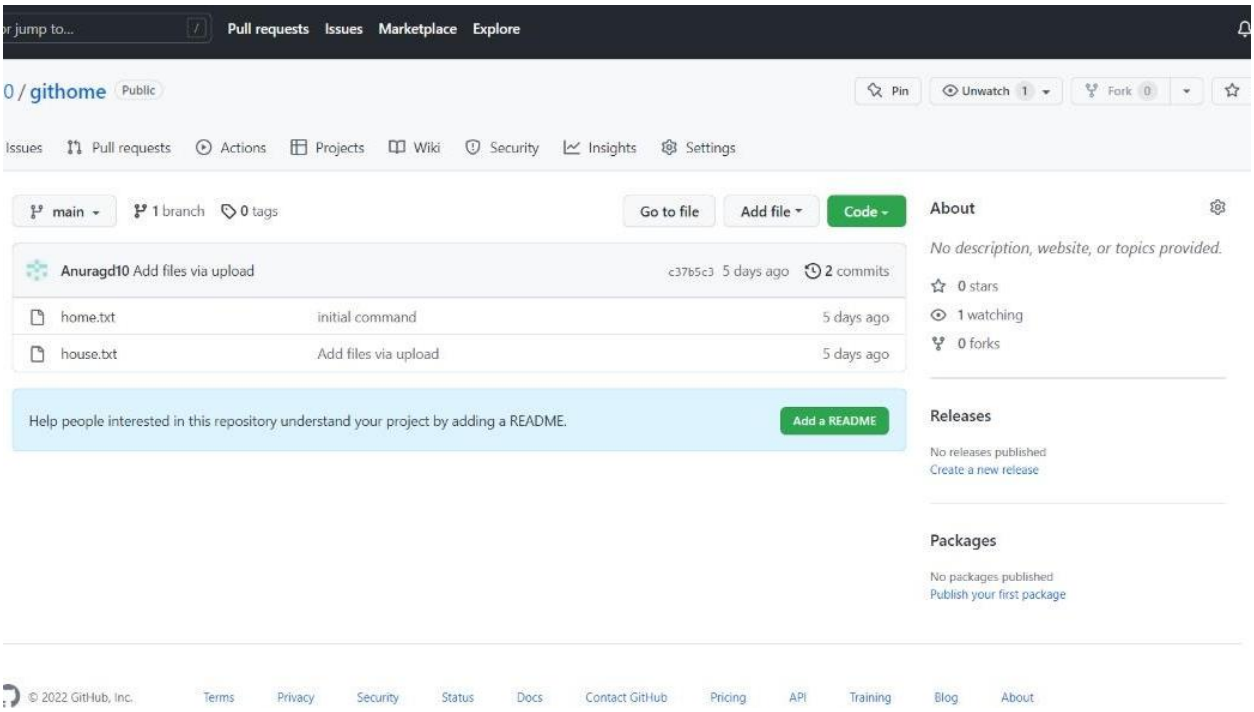
```
MY PC@DESKTOP-3AJJO6R MINGW64 /e/Git Terminal (master)
$ git commit -m "First Commit"
[master (root-commit) 56a0285] First Commit
 2 files changed, 5 insertions(+)
 create mode 100644 Source Code.docx
 create mode 100644 Steps.txt

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
```

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

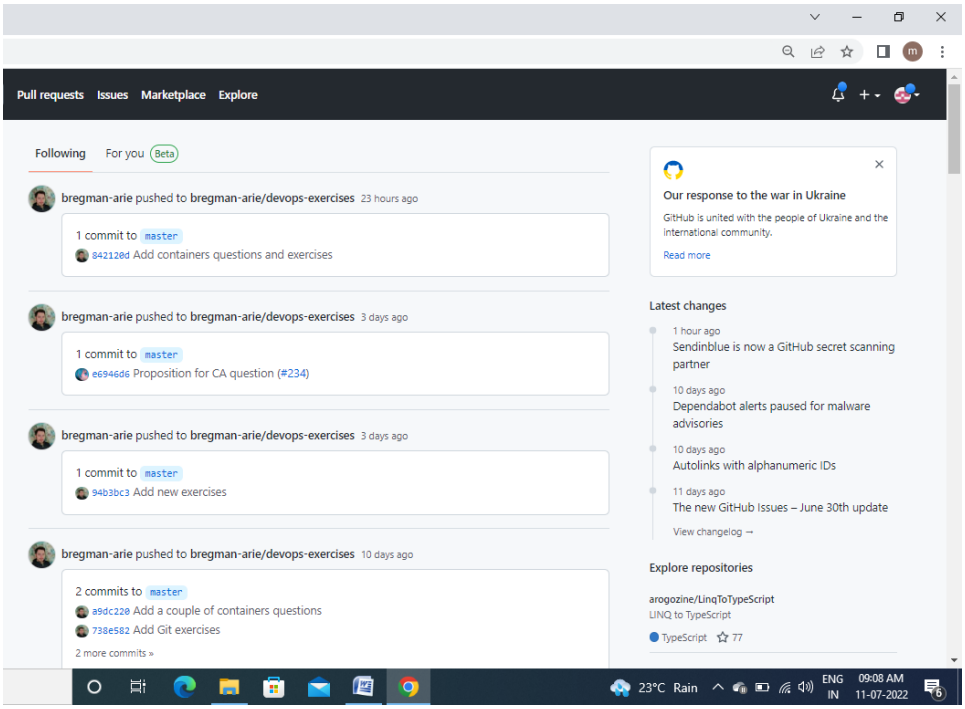


Now we can see that the files we want to push are updated here successfully.

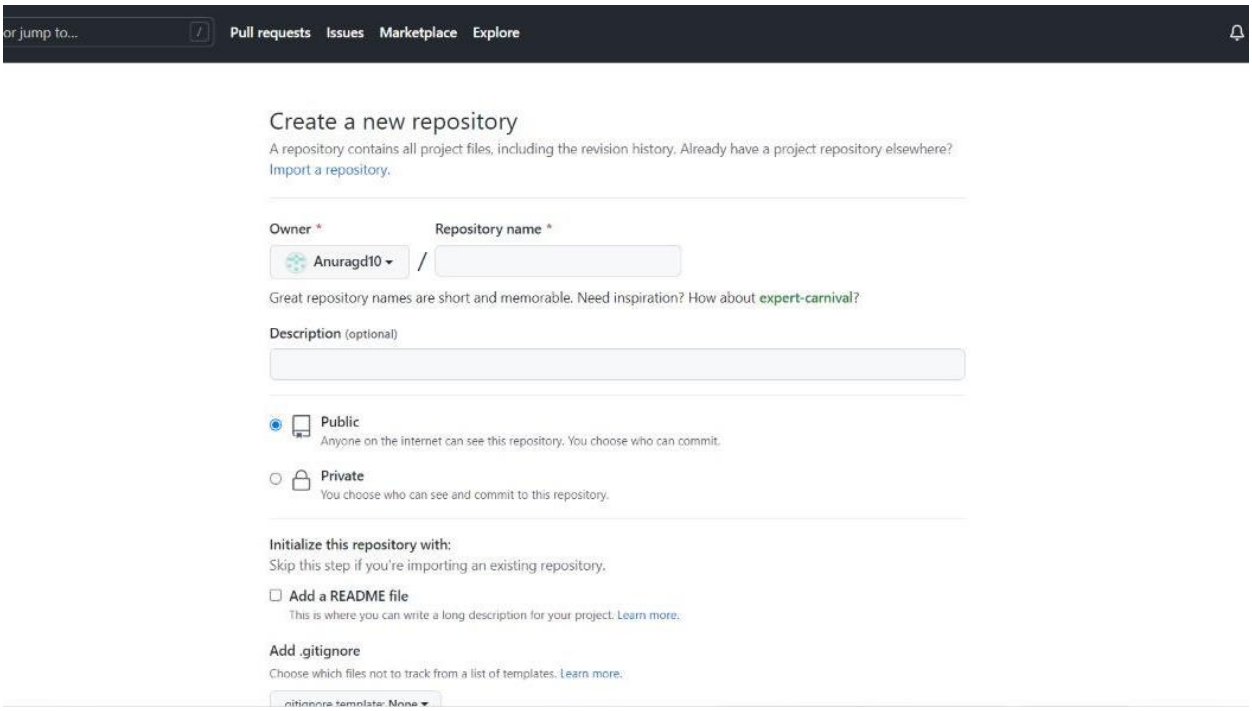
5. Initializing a Local repository using Git GUI

.

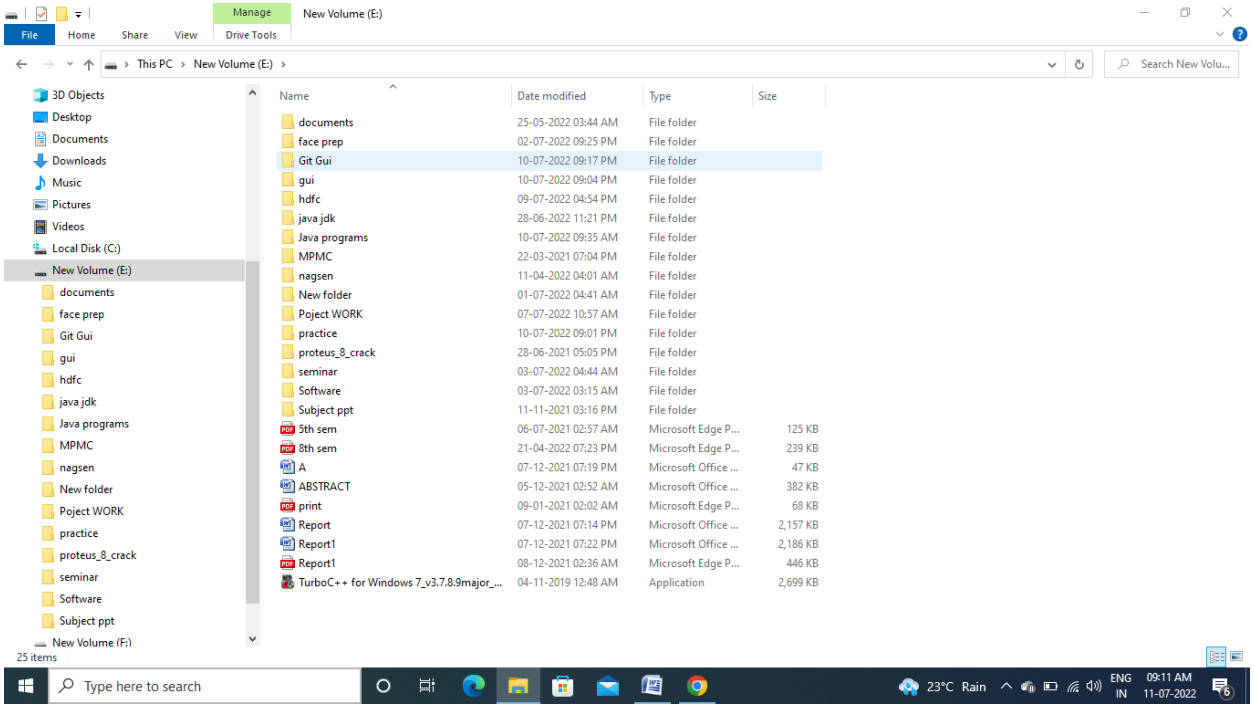
Step 1: Sign in in GitHub account. Click on the plus icon available on the top right corner. Make a repository on Github.



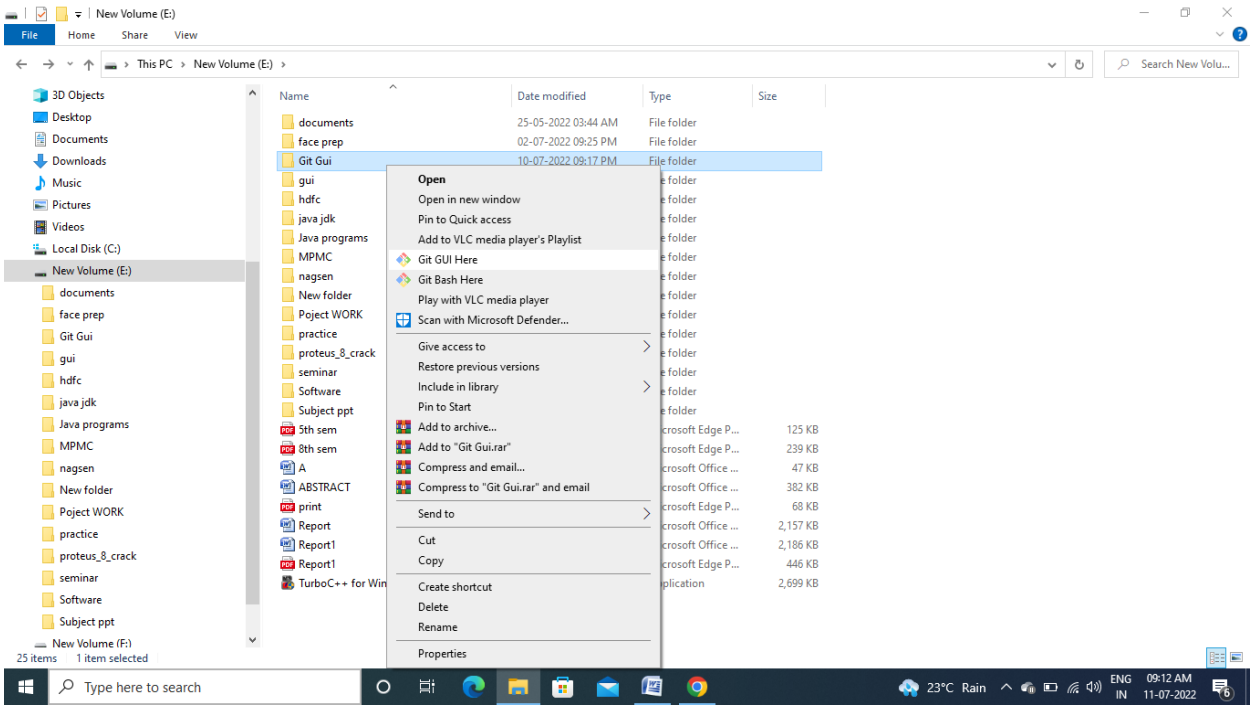
Step 2: Give a suitable name to your repository and click on create repository.



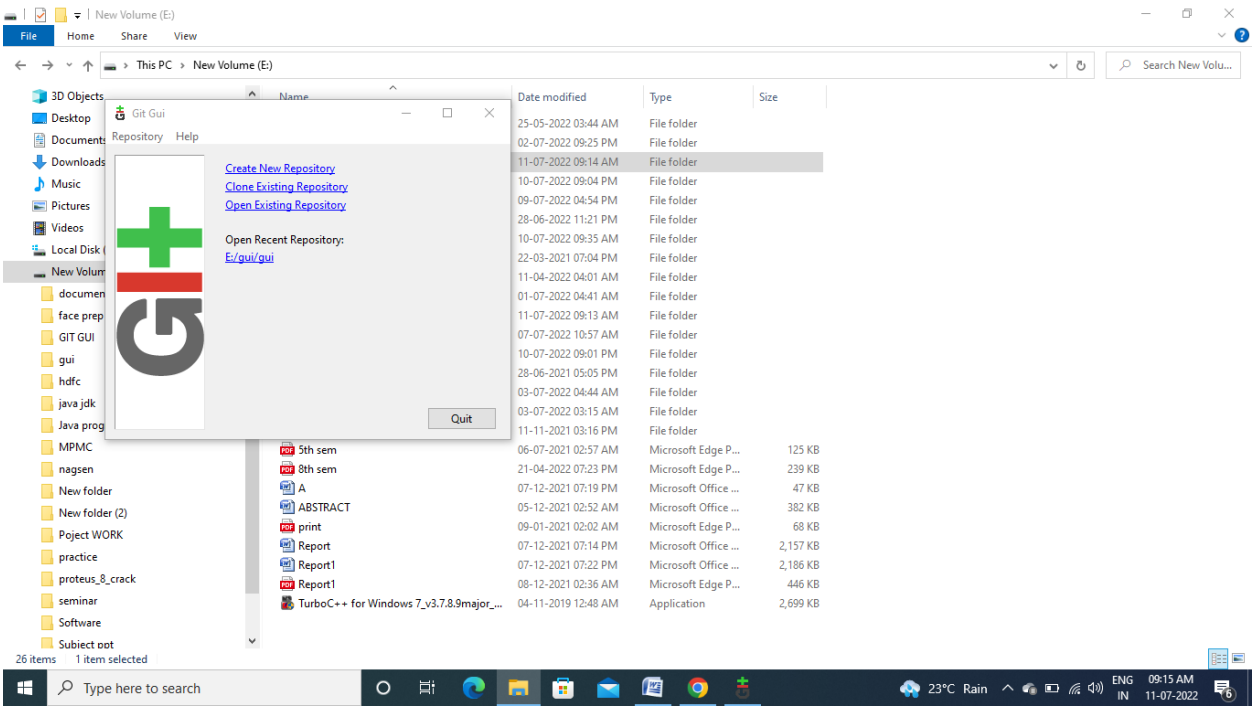
Step 3: Make any folder at any drive in your system and then right click.



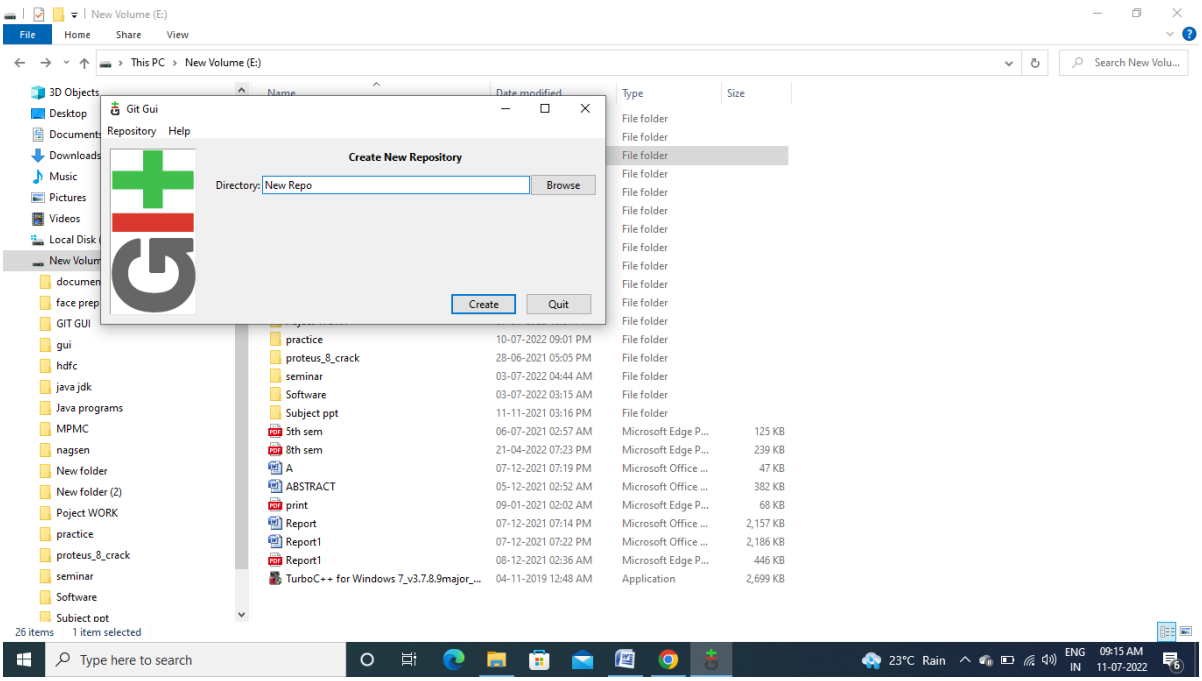
Step 4: Right click on created folder and click Git GUI here.



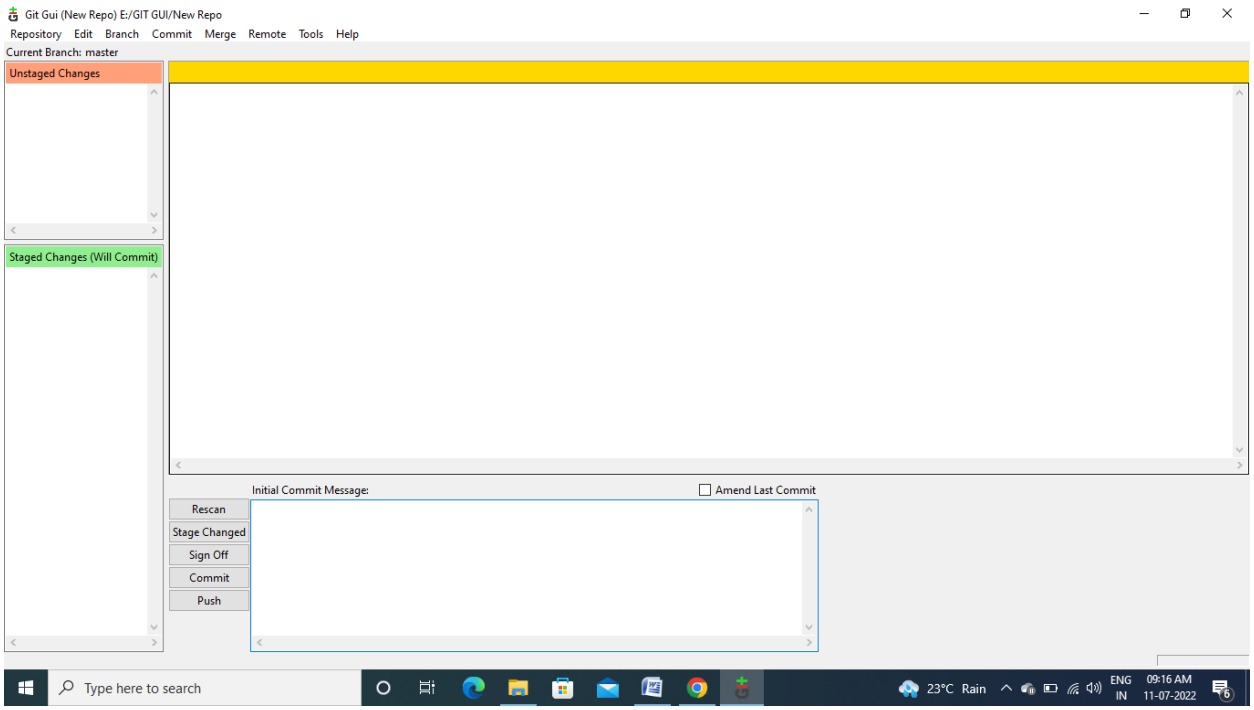
Step 5: Click on Create New Repository


Git Gui
Repository Help
Create New Repository
Clone Existing Repository
Open Existing Repository
Open Recent Repository:
E:/gui/gui
Quit
Type here to search

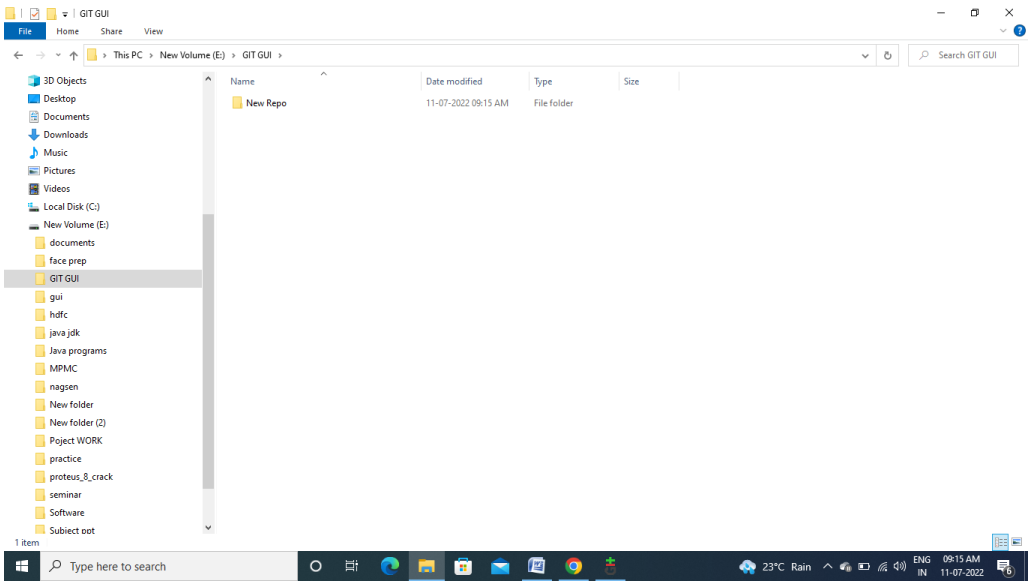
Step 6: Give the suitable name to the Dicrectory and click on Create.



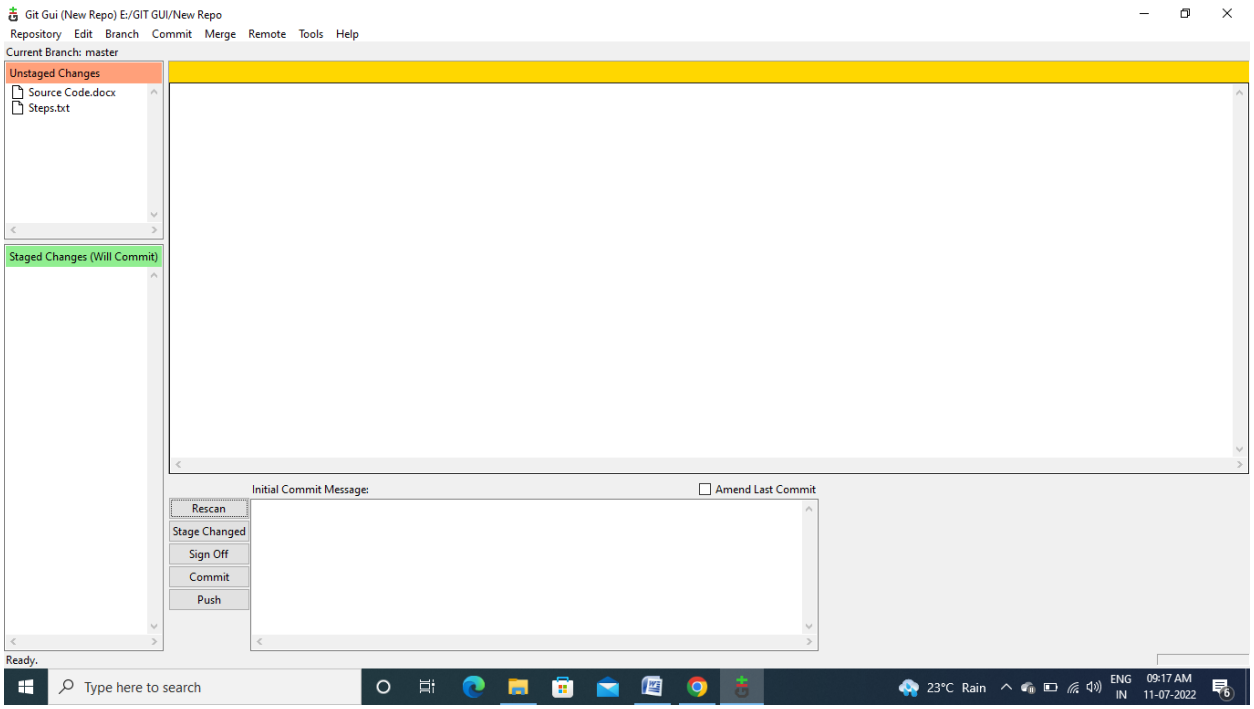
Step 7: After creating the directory this interface will open.



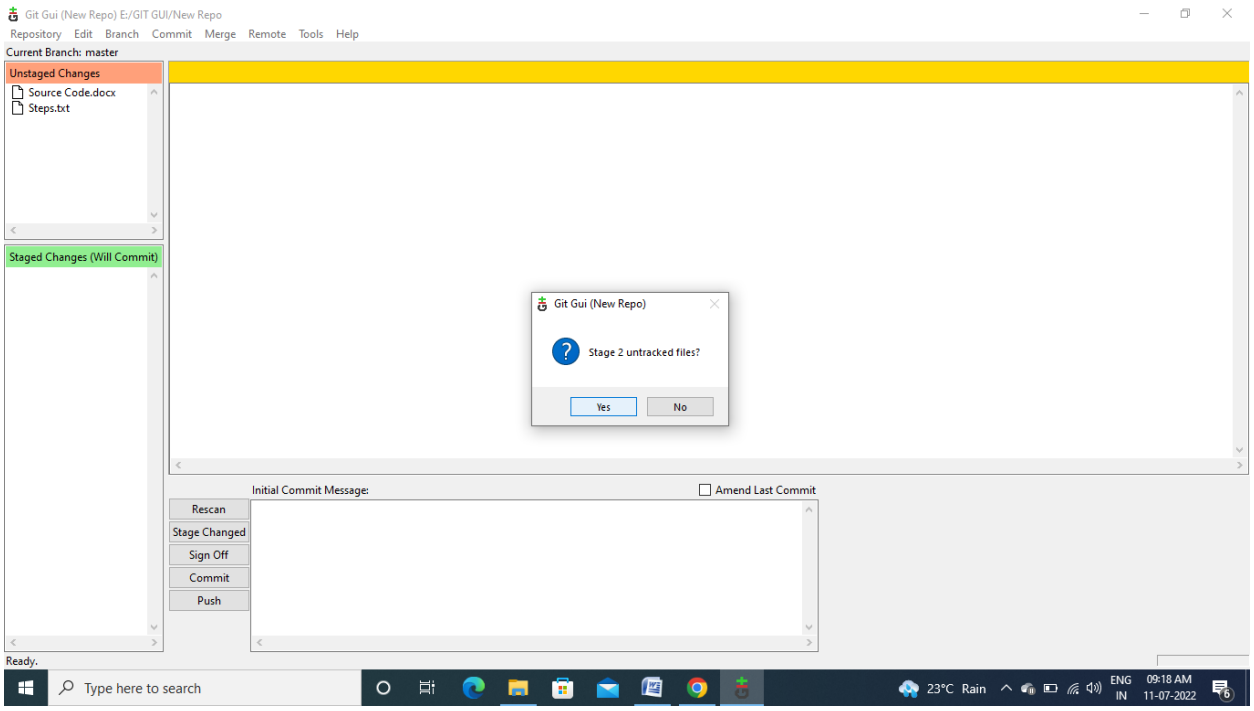
Step 8: Now the new directory is created. Now add your project/work/ files to this folder.



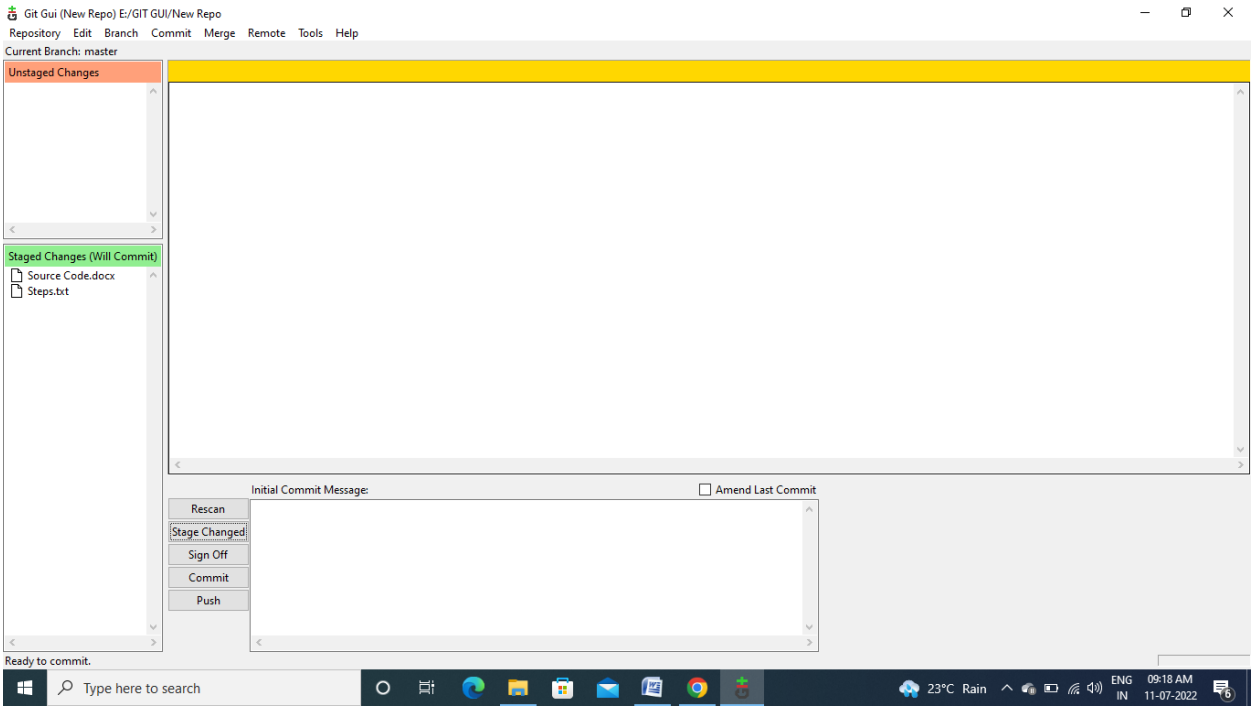
Step 9: Go to Git GUI and click on Rescan button. Now the files are seen in Unstaged Changes section.



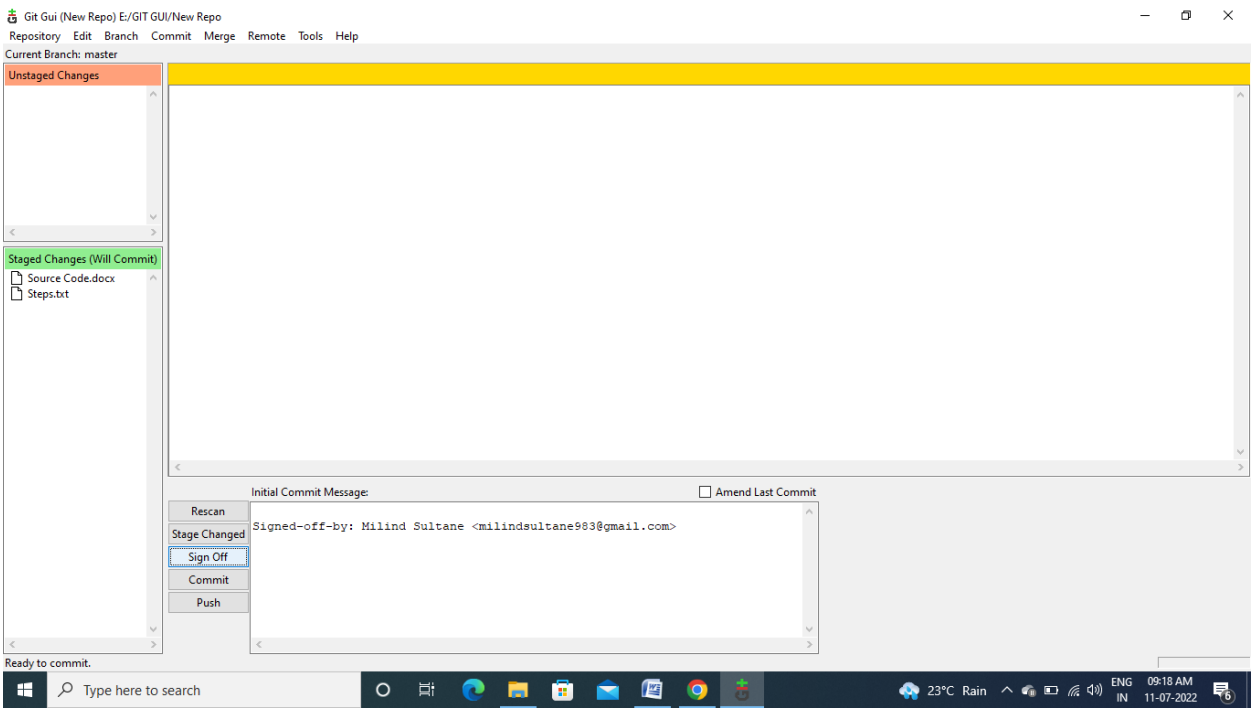
Step 10: To push the files we have to move our files from Unstaged Changes to the Staged Changes section. Click on Stage Changed button & click Ok to the pop up.


Git Gui (New Repo) E:/GIT GUI/New Repo
Repository Edit Branch Commit Merge Remote Tools Help
Current Branch: master
Unstaged Changes
Source Code.docx
Steps.txt
Staged Changes (Will Commit)
Git Gui (New Repo)
Stage 2 untracked files?
Yes
No
Initial Commit Message:
Amend Last Commit
Rescan
Stage Changed
Sign Off
Commit
Push
Ready.
Type here to search
23°C Rain
ENG
IN
09:18 AM
11-07-2022

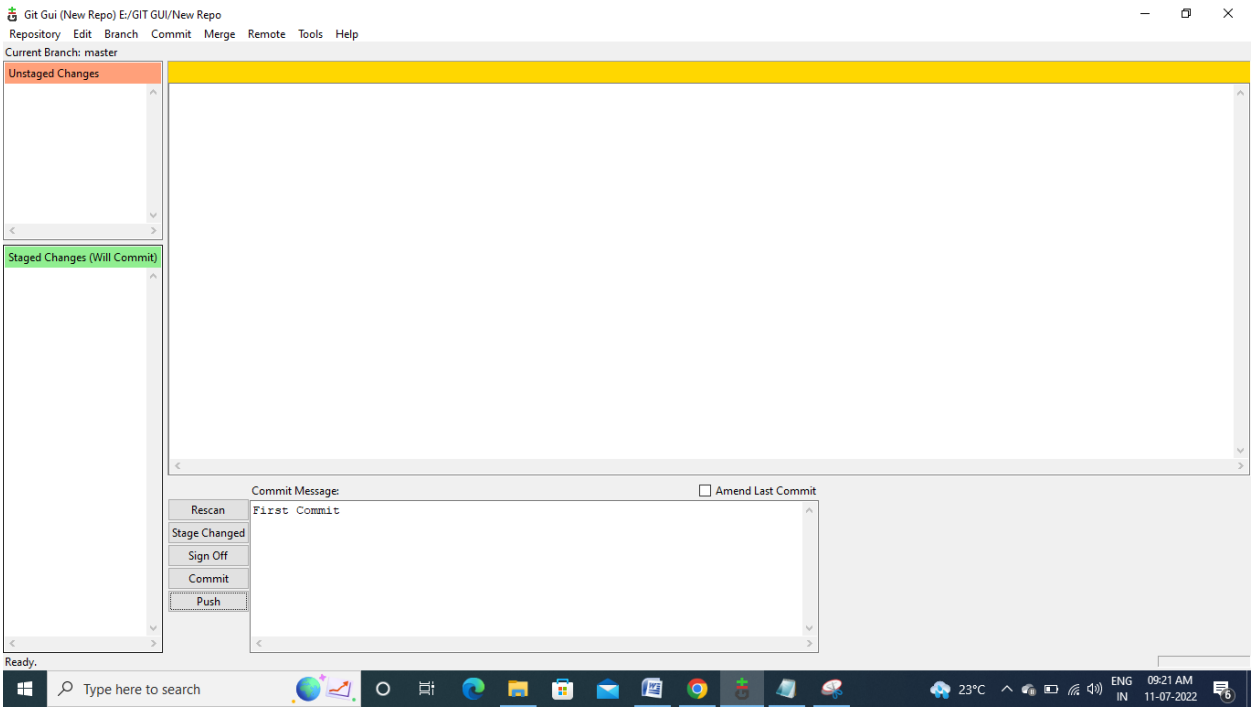
Step 11: Then we can see here files moved from Unstaged Changes to Staged Changes



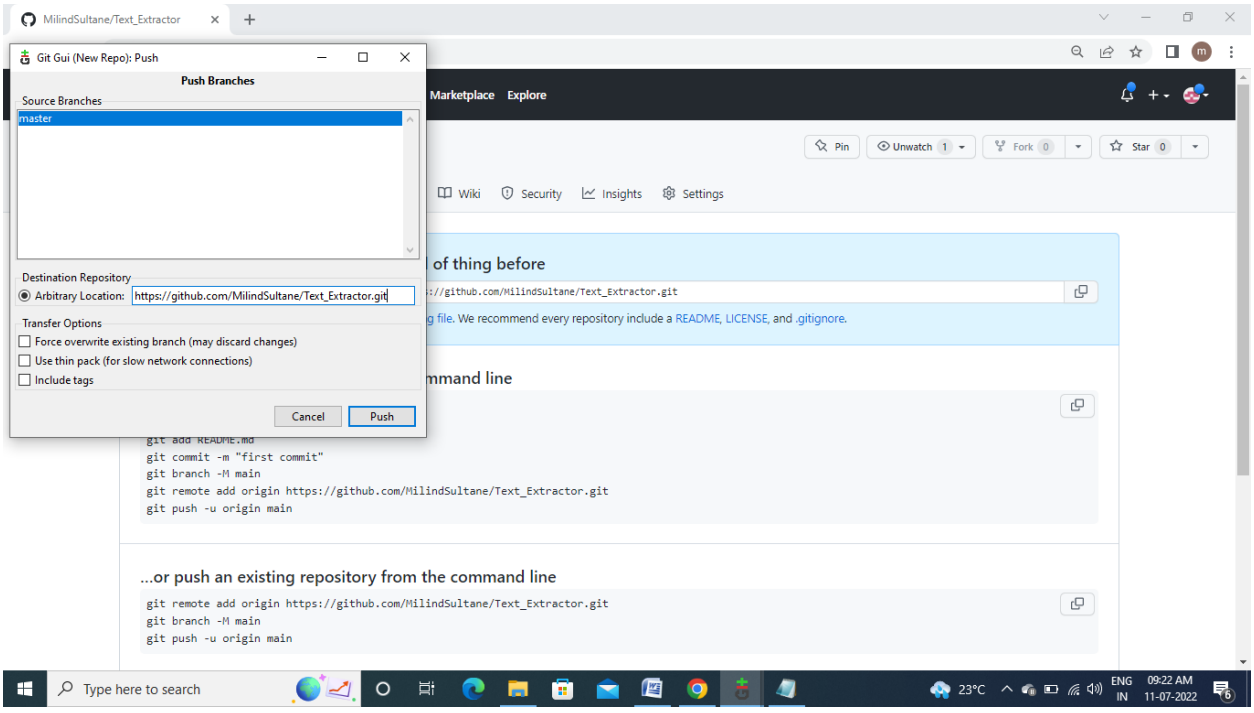
Step 12: Sign off shows the information about the user who commit the changes. It will shows the name and email address of the user.



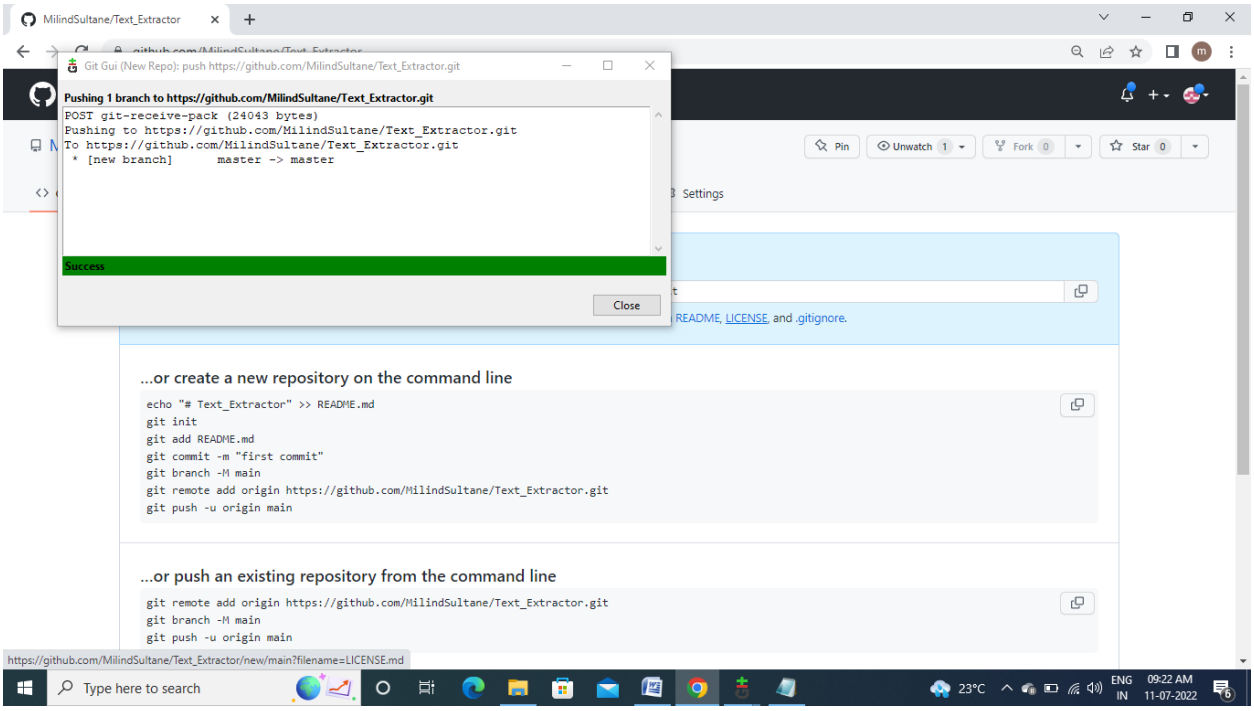
Step 13: Commit whatever you want to commit and then click on Commit button.


Git Gui (New Repo) E:/GIT GUI/New Repo
Repository Edit Branch Commit Merge Remote Tools Help
Current Branch: master
Unstaged Changes
Staged Changes (Will Commit)
Commit Message:
Amend Last Commit
Rescan
Stage Changed
Sign Off
Commit
Push
First Commit
Ready.
Type here to search
23°C
ENG
IN
09:21 AM
11-07-2022

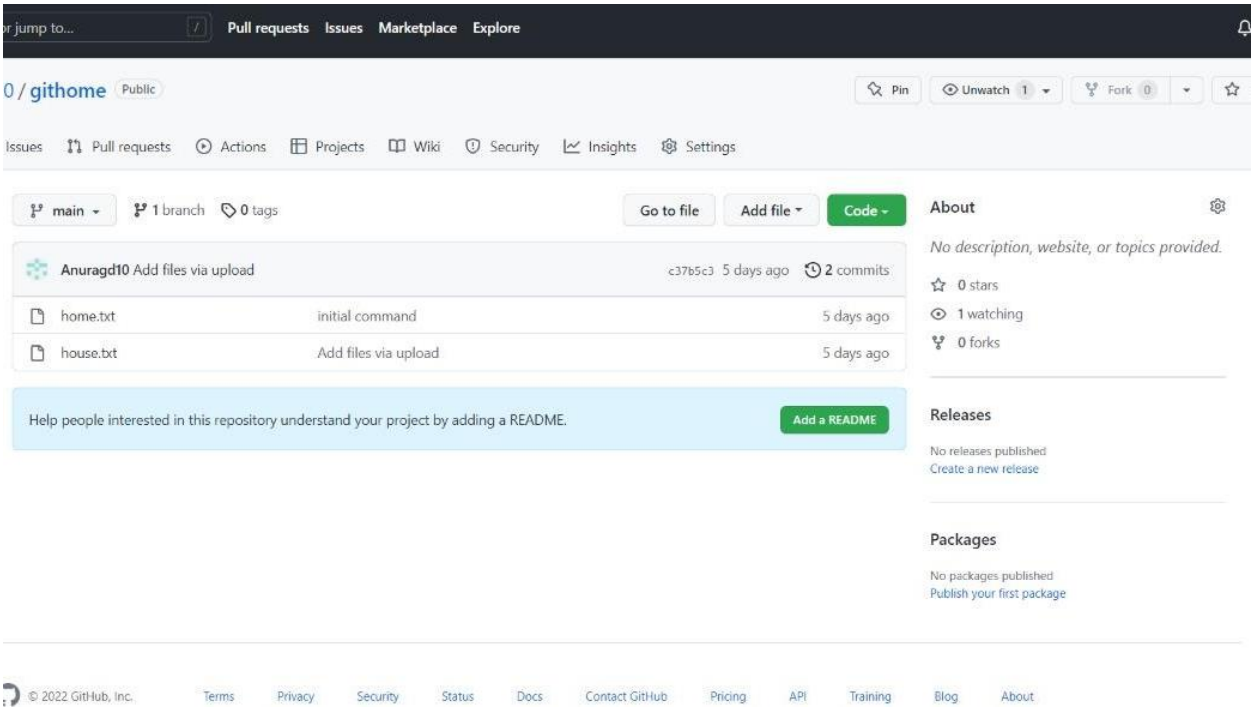
Step 14: Click on Push button. After clicking on Push button it will ask you arbitrary location. Then go to the GitHub copy the https link and paste here



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.


jump to...
Pull requests
Issues
Marketplace
Explore
0 / githome
Public
Pin
Unwatch 1
Fork 0
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
About
No description, website, or topics provided.
0 stars
1 watching
0 forks
Anuragd10 Add files via upload
c37b5c3 5 days ago
2 commits
home.txt
initial command
5 days ago
house.txt
Add files via upload
5 days ago
Help people interested in this repository understand your project by adding a README.
Add a README
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
© 2022 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About