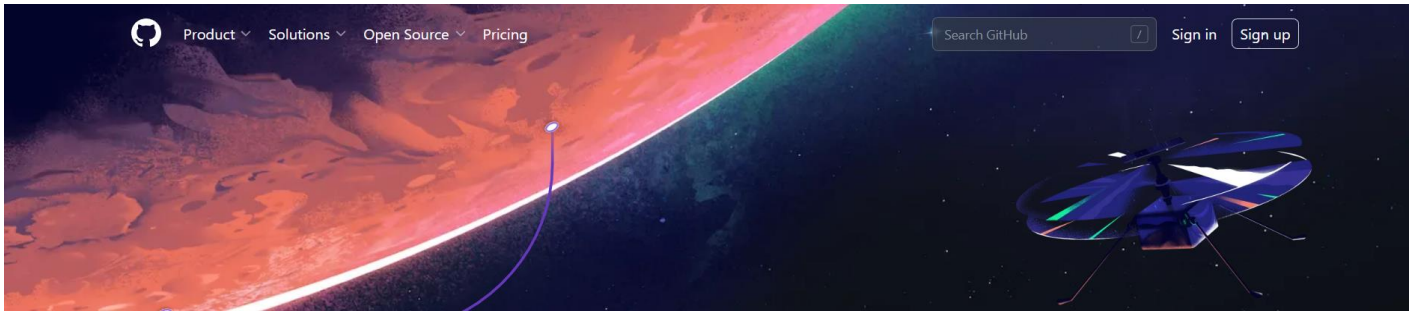
ASSIGNMENT-8

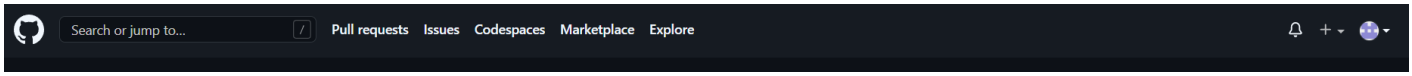
Problem Statement: Deploy a project from Local Machine to GitHub and vice-versa.

Procedure:

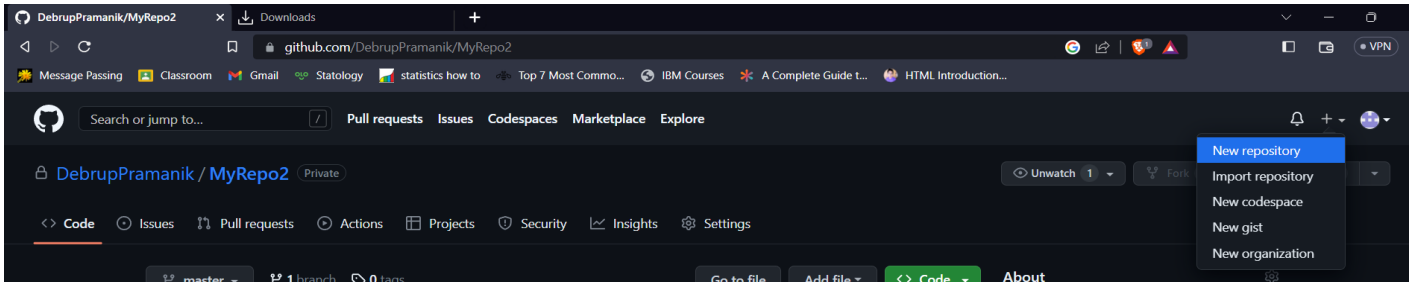
1. Make sure you have installed Git for Windows Application if you are using a Windows machine. If not then download the same from https://gitforwindows.org/ website. Install and make sure you have integrated it with the Windows Shell. (It is checked on by default so no need to change anything.)
2. Now go to GitHub Website https://github.com/ and Sign In to your account.
   If you do not have an account then click on Sign Up button to Create an account for yourself. Then just follow the on-screen instructions to complete your registration. Now finally you will Sign-In to your account.



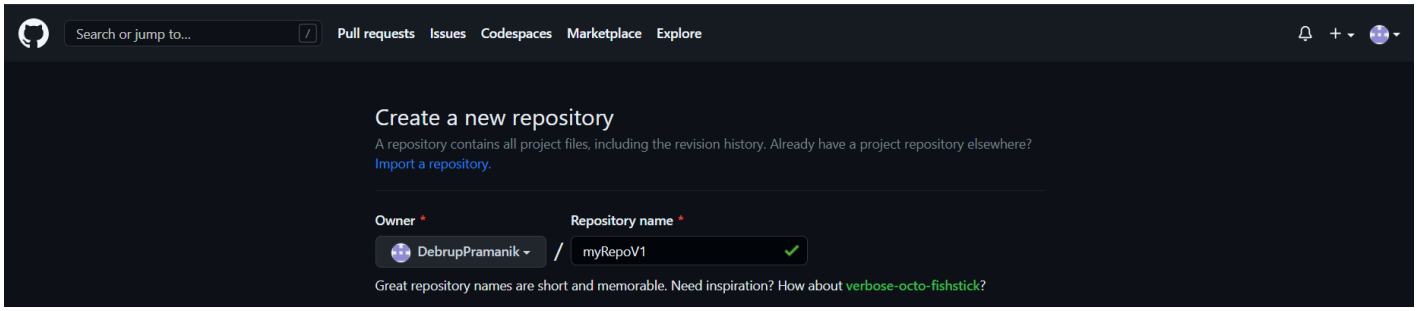
3. After successfully signing in GitHub, click on the + button situated on the top right corner of the website beside your profile picture.



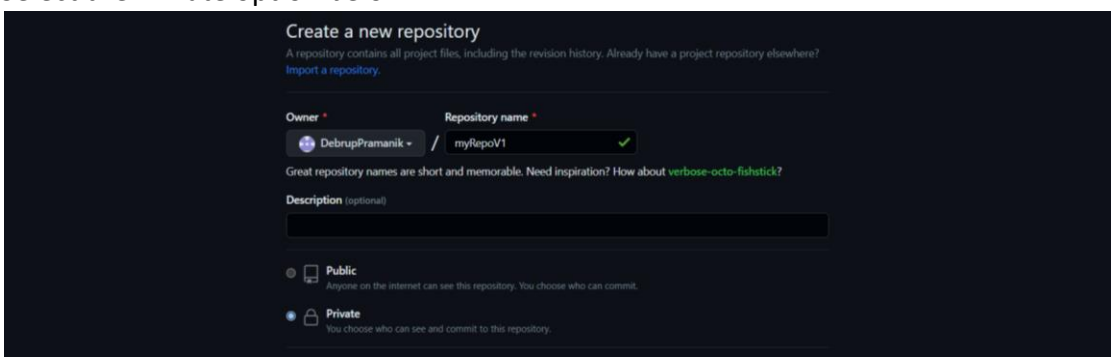
4. Now after clicking a menu will appear. Click on New Repository. This will create a New Repository where you can upload/deploy your project folders and files.



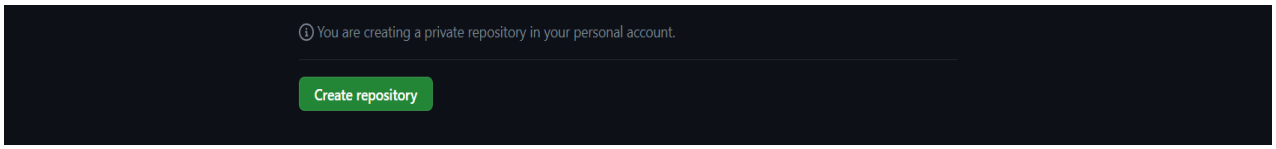
5. Now Enter the name of the Repository.



6. Select the Private option below.



7. Next scroll-down and click on Create Repository.

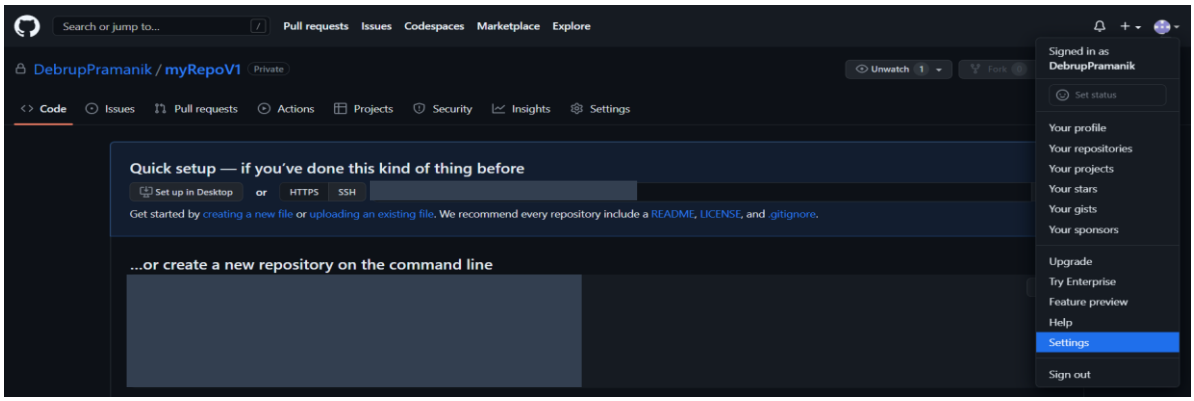


8. You will be redirected to the Repository code page.
Now while adding your project to your GitHub repository, you need to provide credentials of your account and sign-in every time you are doing this, through the Git bash terminal. Or there is another way, and that is by using Tokens generated for your account.

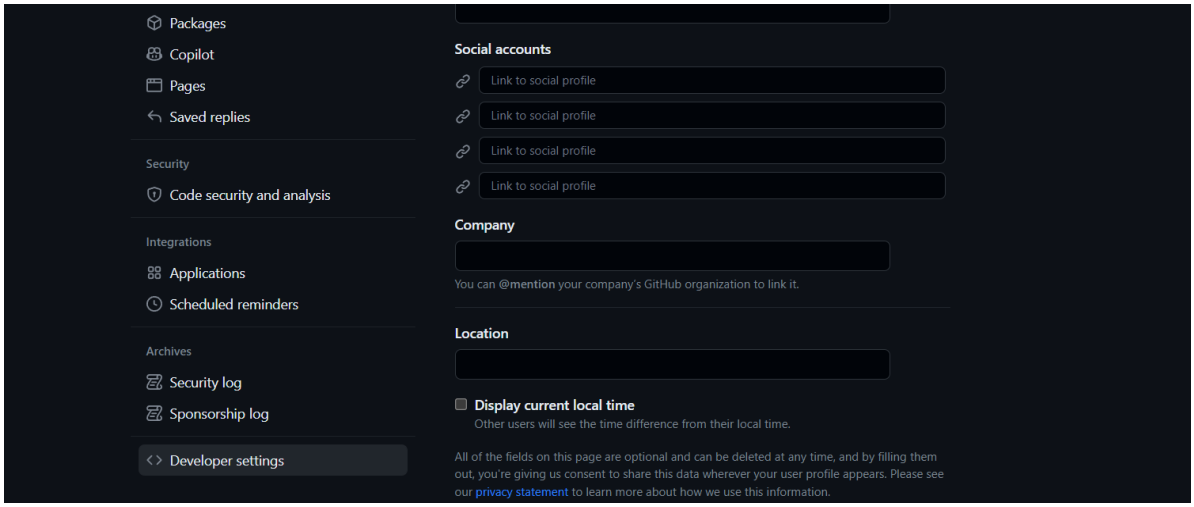
9. So, for generating token for your account follow the steps:

a. Click on your profile on the top right corner of the web page.

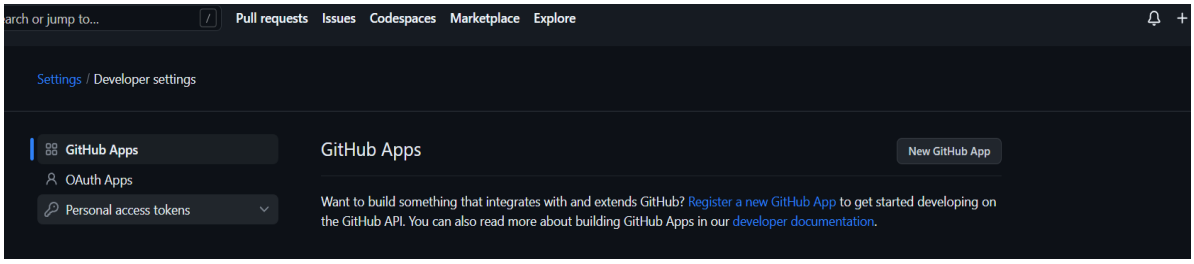
b. A drop-down menu will appear. Click on the Settings option.



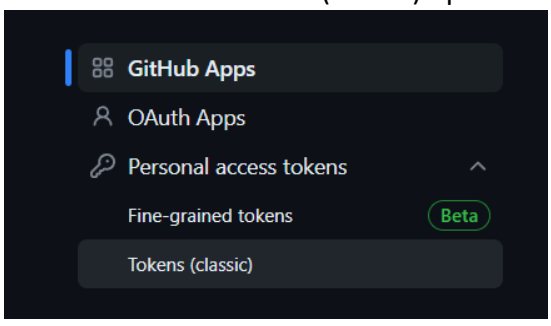
c. Now scroll down until you see a Developer Options button on the left Nav bar. Click on it.



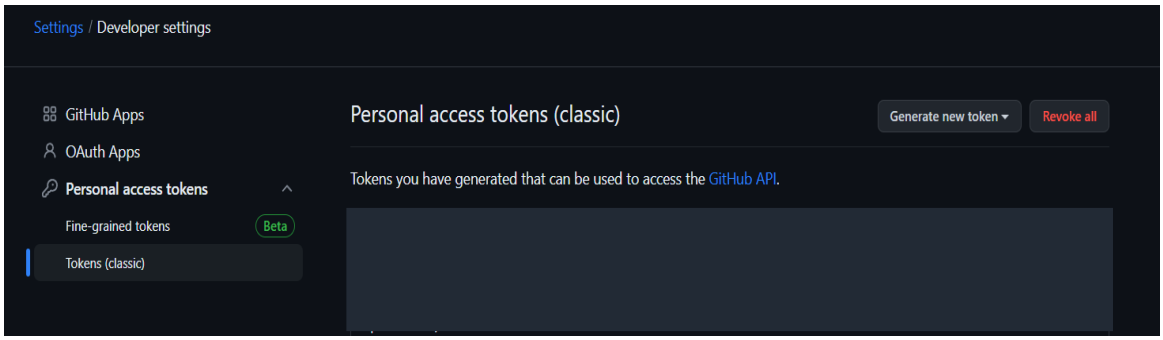
d. Now click on the Down arrow beside Personal Access Tokens in the left Nav bar.



e. Next click on the Tokens(classic) option.

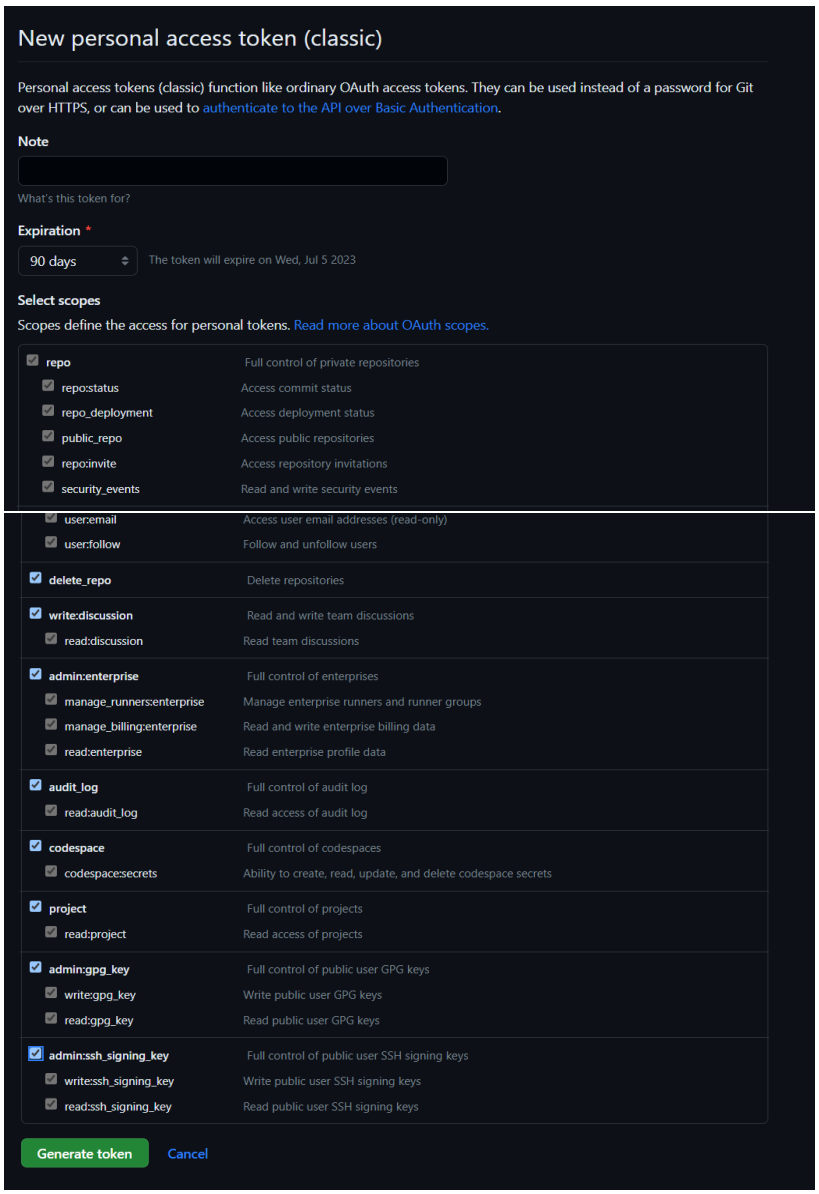


f. Next click on Generate New Token button



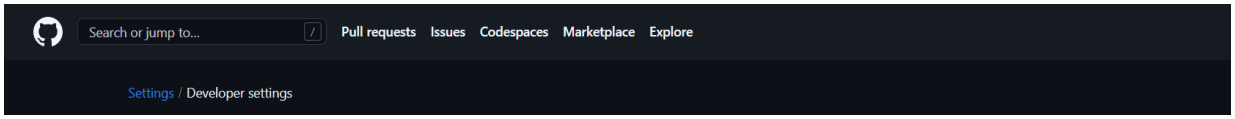
g. Again, click on Generate New Token(classic).

h. Give the Token Name.
Select the Expiration time (Preferably 90 Days).
Now check all the parent boxes below.
Then Click on Generate Token Button.

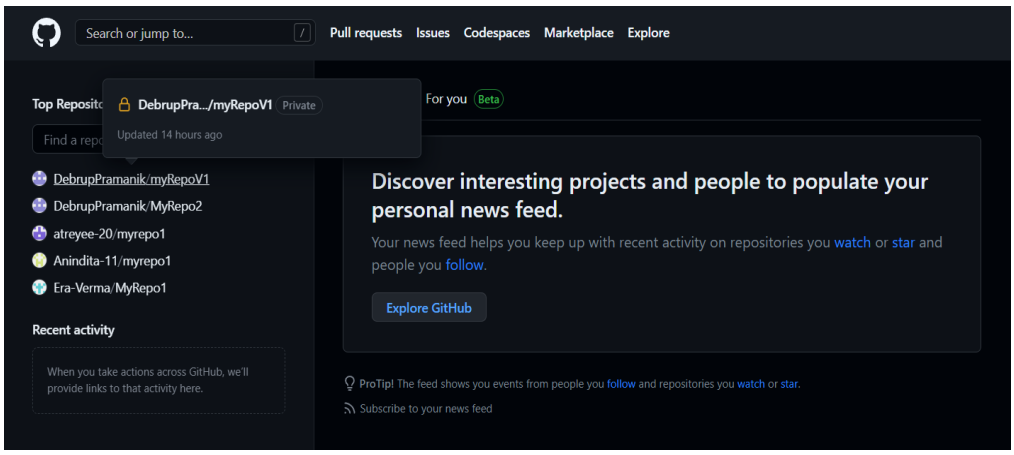


i. Now a Token will be generated. Copy it and save it in a text file. Keep it somewhere safe as we will need it later.

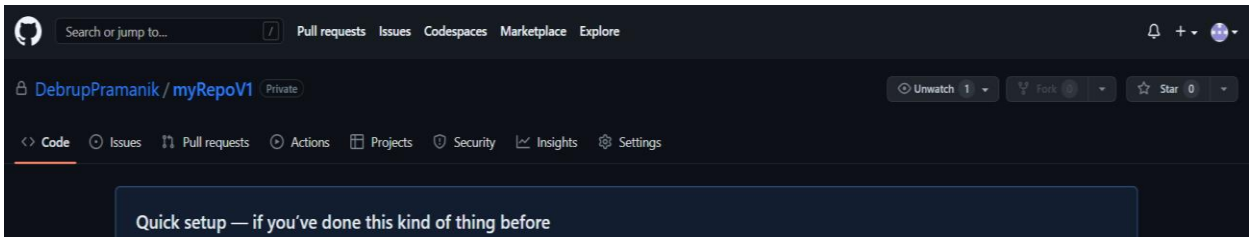
10. Now scroll-up and click on the icon on the top left corner of the web page. This will redirect you to your home page.



11. Now on the left nav bar you will find your Top Repositories You are currently working/contributing to. You can also locate your newly created Repository here. Click on it.

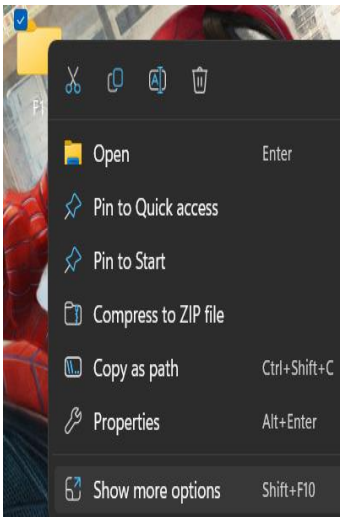


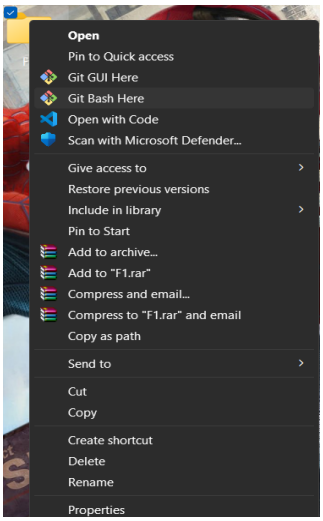
12. Go to the code section and keep it open. Now minimize your browser.



13. Now create a folder anywhere in your computer. Give it a name. Now Right Click on it and select Git Bash here.

If you are in Win 11 then after Right click go to Show more Options and then select Git Bash here.



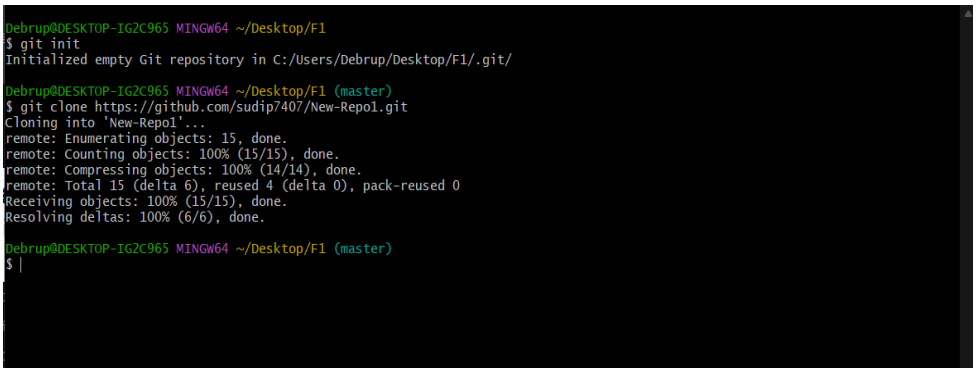


14. It will open the Git Bash Terminal

15. Now type the following to **<u>clone</u>** the project provided/required (Link will be provided or specified or can be obtained from visiting the required project's GitHub page and copying it):

➔ **git init**

➔ **git clone** https://github.com/sudip7407/New-Repo1.git

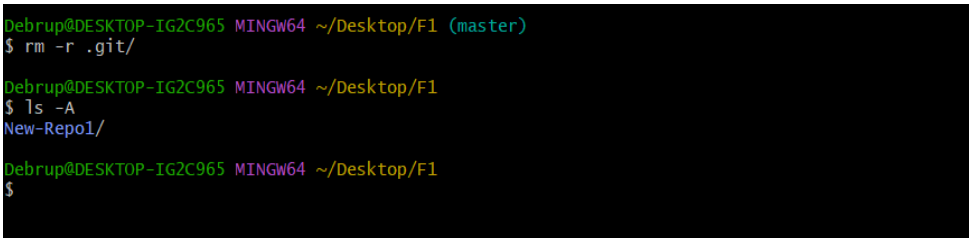


➔ **ls -A**

(We see all the files downloaded from the repository we just cloned. Now our job is to remove/delete the .git/ files from all the folders, including nested ones. Also, we have to remove. gitignore files from wherever it is present.)

➔ **rm -r .git/**

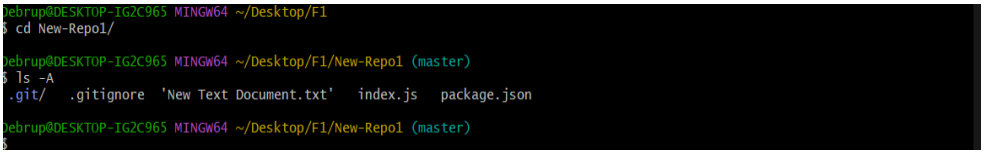
➔ **ls -A**



(We have successfully removed the .git/ file from the main directory. However we have to check the New-Repo1/ directory to see if it has no unwanted files.)

➔ **cd New-Repo1/**

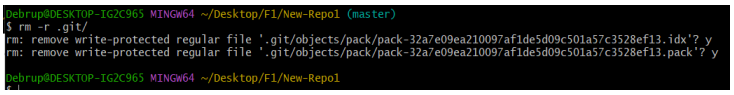
➔ **ls -A**



(We now see we have to remove the .git/ and .gitignore files. Then we need to go back to the main directory)

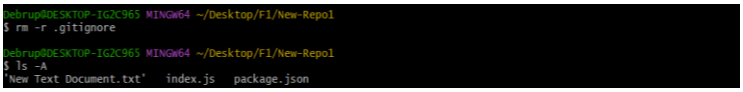
(Sometimes permissions are asked when removing some files. Just type y when prompted and press enter and repeat it when asked every time.)

➔ **rm -r .git/**



➔ **rm -r .gitignore**

➔ **ls -A**



(Our work here is done. Now we need to return to our main directory)

➔ **cd ..**

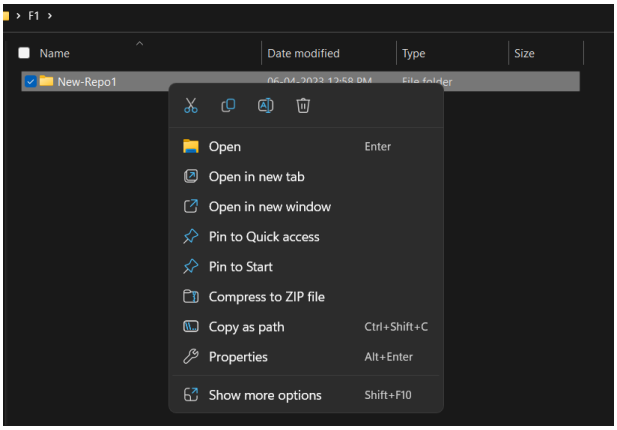
```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1
$ cd ..

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1
$ ls -A
New-Repo1/
```

We now have successfully cloned our required project.

16. Now close the terminal and open the folder in which you just cloned the project.

17. Single Click on the New-Repo1 file and right click and just like explained above again select Git bash here option.



18. Again, a Git bash terminal will open.

19. Again, type the following commands but now we will now upload this cloned project to our created repository on our GitHub:

➔ **git init**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1
$ git init
Initialized empty Git repository in C:/Users/Debrup/Desktop/F1/New-Repo1/.git/

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$
```

➔ **ls -A**

➔ **git status**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1
$ git init
Initialized empty Git repository in C:/Users/Debrup/Desktop/F1/New-Repo1/.git/

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ ls -A
 .git/  'New Text Document.txt'   index.js   package.json

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        New Text Document.txt
        index.js
        package.json

nothing added to commit but untracked files present (use "git add" to track)
```

➔ **git config --global user.email “Your email here”**

➔ **git config --global user.name “Your GitHub account username here”**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git config --global user.email "debrup202002@gmail.com"

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git config --global user.name "DebrupPramanik"
```

➔ **git config user.name**

➔ **git config user.email**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git config user.name
DebrupPramanik

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git config user.email
debrup202002@gmail.com

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
```

(this is required to set as it will commit the files in your repository using the set username and email.)

➔ **git add .**

➔ **git status**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git add .
warning: in the working copy of 'cls', LF will be replaced by CRLF the next time
 Git touches it

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   New Text Document.txt
        new file:   cls
        new file:   index.js
        new file:   package.json


Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$
```

(Notice all the files that was colored red previously is now green, indicating that they have been added to your local repository. We now have to commit the changes we made and then push it to our GitHub repository)

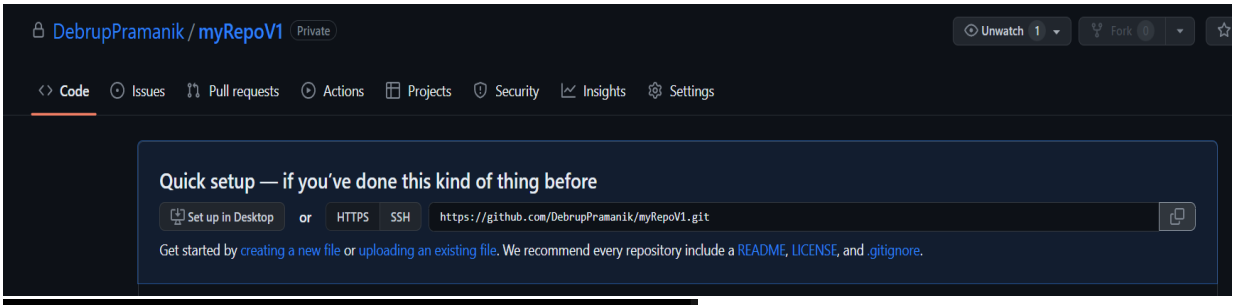
➔ **git commit -m “type your own message here”**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git commit -m "Hello World. My first ever commit"
[master (root-commit) 36046dc] Hello World. My first ever commit
 4 files changed, 49 insertions(+)
 create mode 100644 New Text Document.txt
 create mode 100644 cls
 create mode 100644 index.js
 create mode 100644 package.json

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$
```

➔ **git remote add origin https://github.com/yourusername/Yourrepositoryname.git**

(The https address is the address of your repository. To obtain it maximize your browser where your GitHub repository is open and there copy the https address)



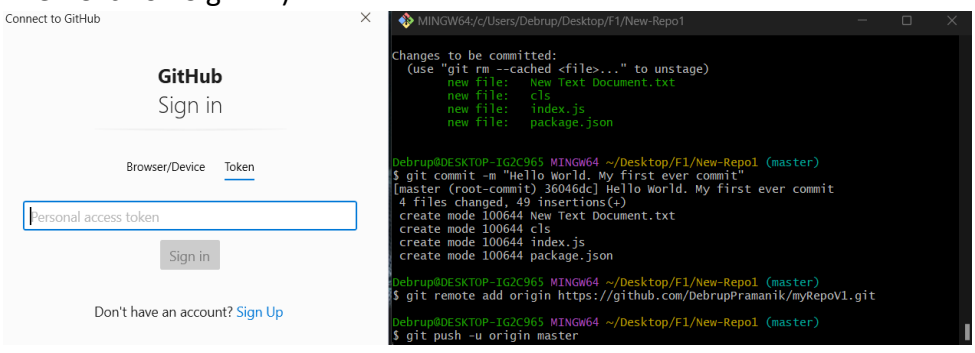

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git commit -m "Hello World. My first ever commit"
[master (root-commit) 36046dc] Hello World. My first ever commit
 4 files changed, 49 insertions(+)
 create mode 100644 New Text Document.txt
 create mode 100644 cls
 create mode 100644 index.js
 create mode 100644 package.json

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git remote add origin https://github.com/DebrupPramanik/myRepoV1.git

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$
```

➔ **git push -u origin master**

(This is the final command. A pop-up window will open named Connect to GitHub. You will have several options to Sign-In. However, we will be using our Generated token for your account to Sign-In to our account. Click on Token option beside Browser/Device. Paste your token in the placeholder. Then Click on Sign-In.)



(After Successful sign-in the following will show up in the terminal. The Pop-up Window will close automatically)

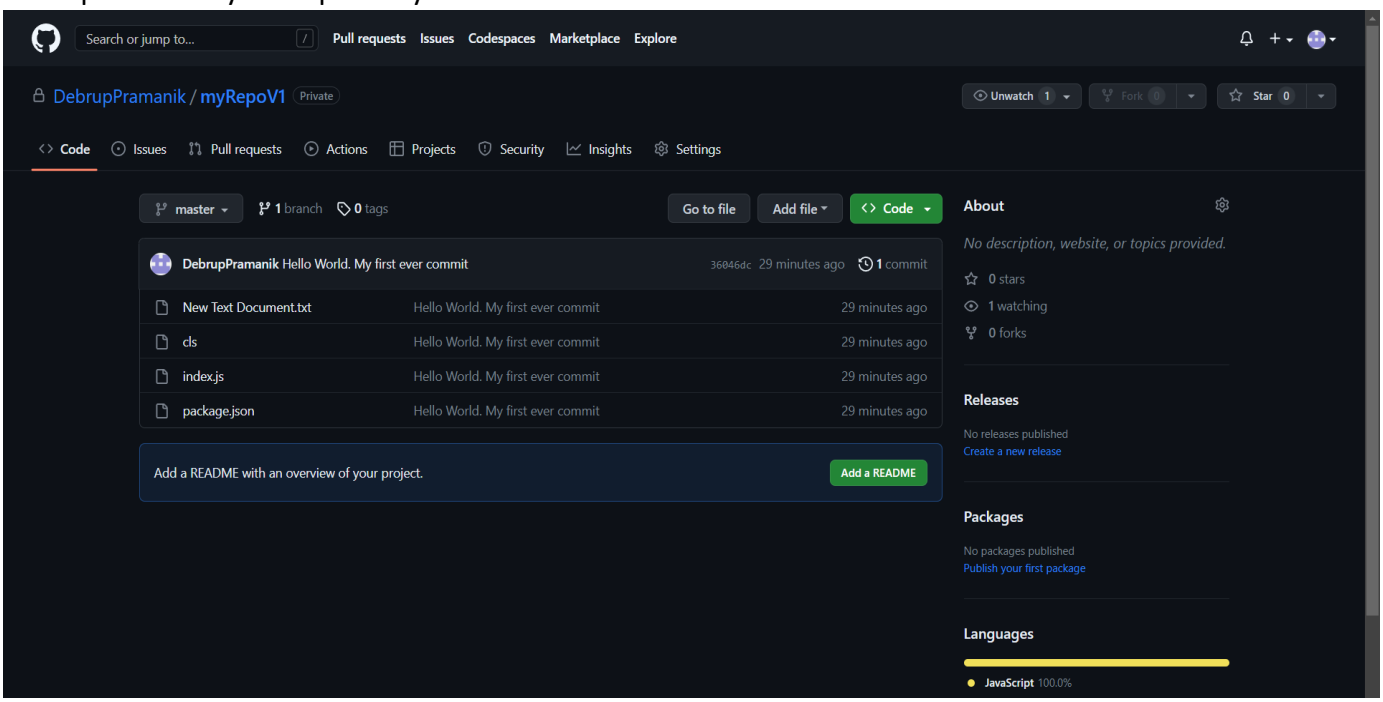
```
[master (root-commit) 36046dc] Hello World. My first ever commit
 4 files changed, 49 insertions(+)
 create mode 100644 New Text Document.txt
 create mode 100644 cls
 create mode 100644 index.js
 create mode 100644 package.json

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git remote add origin https://github.com/DebrupPramanik/myRepoV1.git

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 6 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 1.05 KiB | 1.05 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DebrupPramanik/myRepoV1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$
```

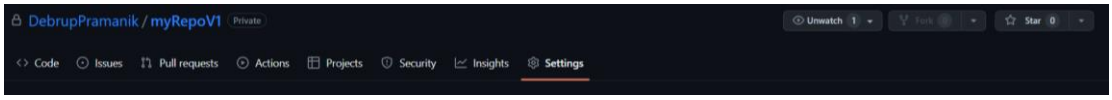
20. Now go to your browser where your GitHub repository is open. Refresh the page. Now you will see the files uploaded in your repository!



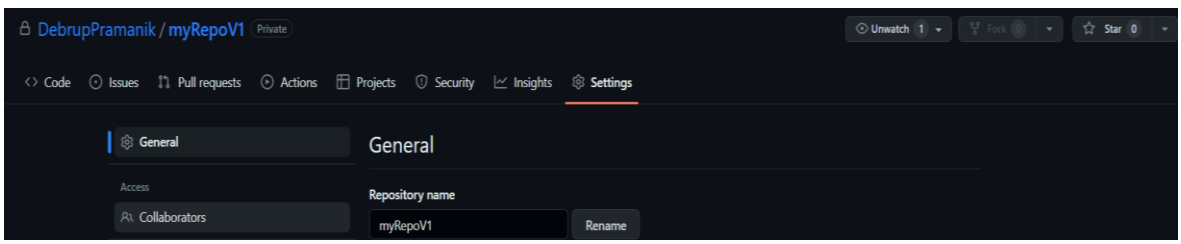
We have successfully Cloned and Uploaded our project to GitHub using Git and Git Bash terminal.

21. You can also add collaborators who can access your private repository and can contribute to it.

a. Go to the Setting section of your repository (It is in the same nav bar as the code section)



b. Then select the collaborators option in the left nav bar.
It will ask you to enter your password.
Enter it.



c. Now you can click on add people button to add others as collaborators of your repository. You have to search by their Username or Email.