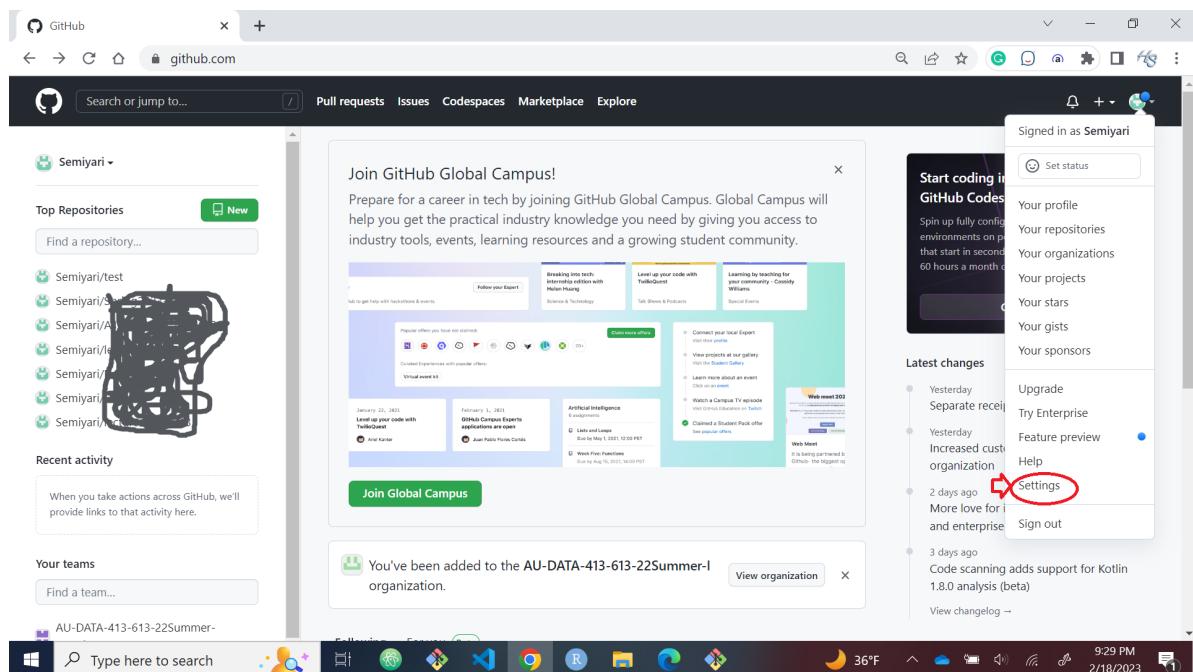


# Push

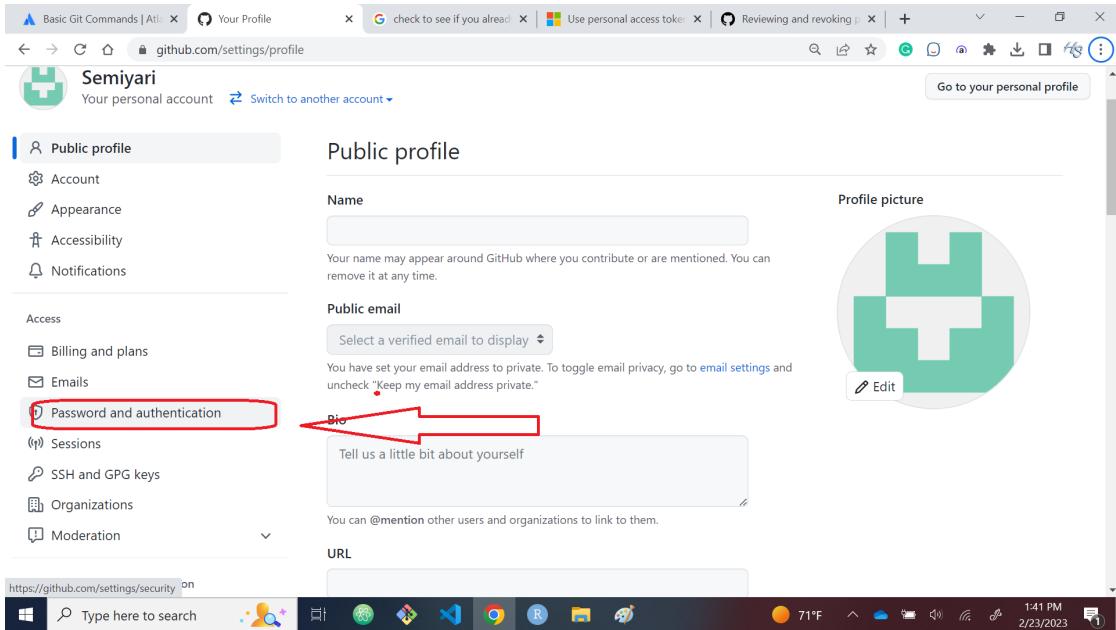
## Check authentication SSH, PAT, and Two-Factors.

### Two-Factors

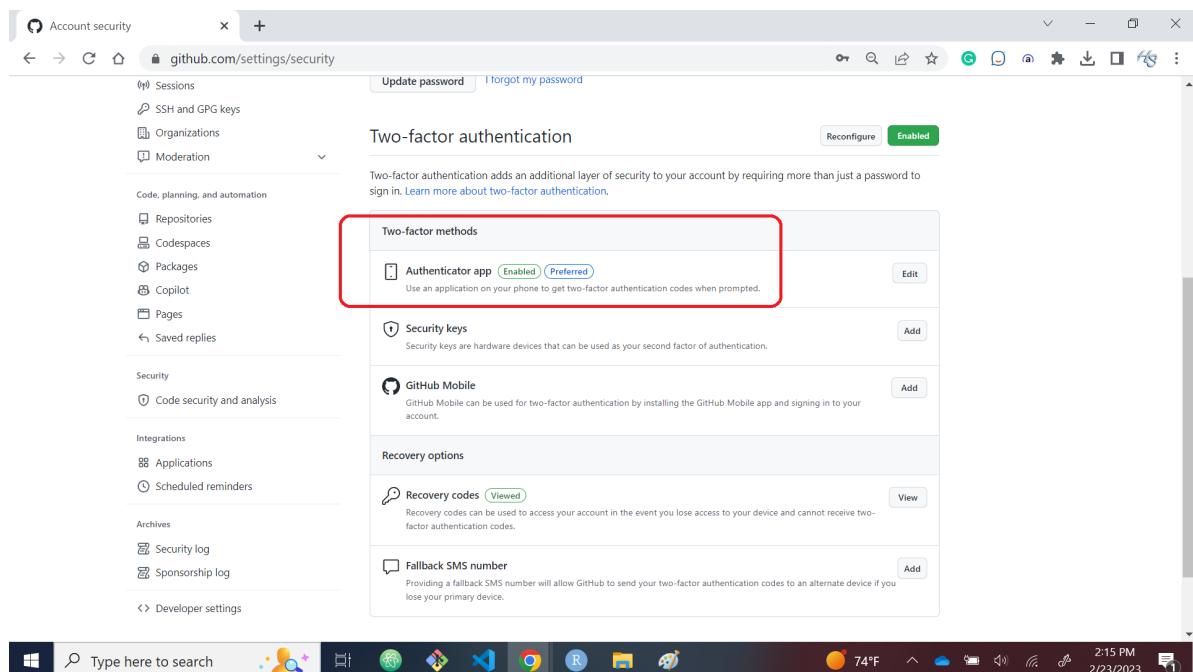
- Step 1. In the upper-right corner of any page, click on your avatar (or your profile photo), then click *Settings*.



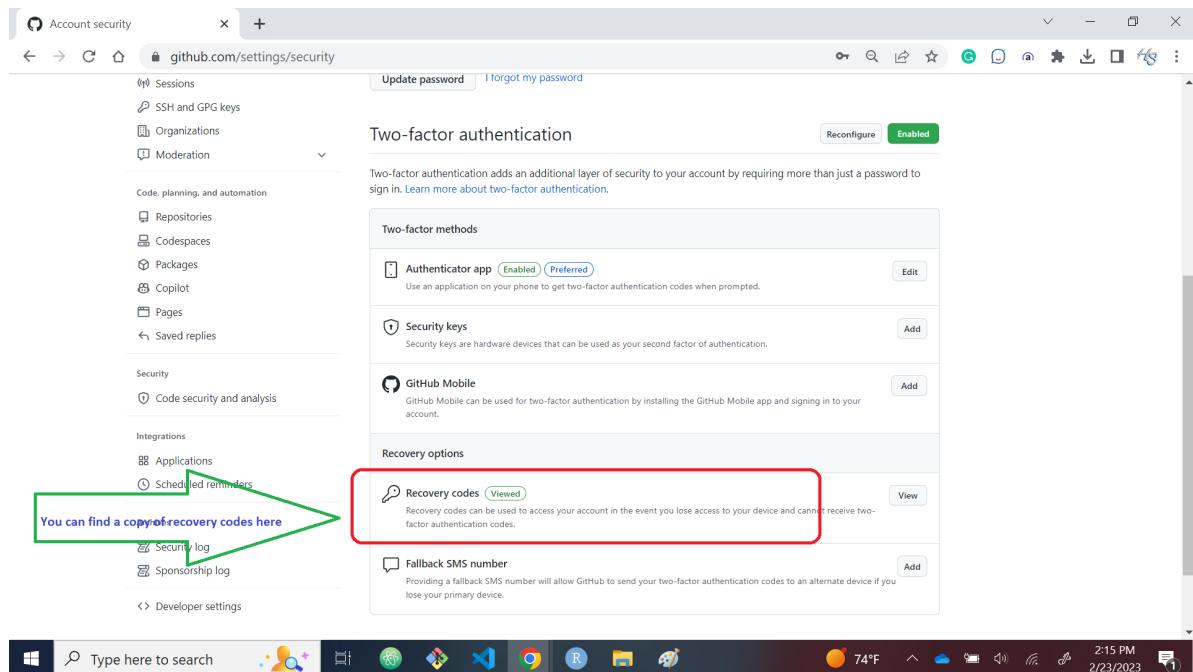
- Step 2. Scroll down and click on “Password and authentication”



- sTEP 3. If you have already set it up you will see the following

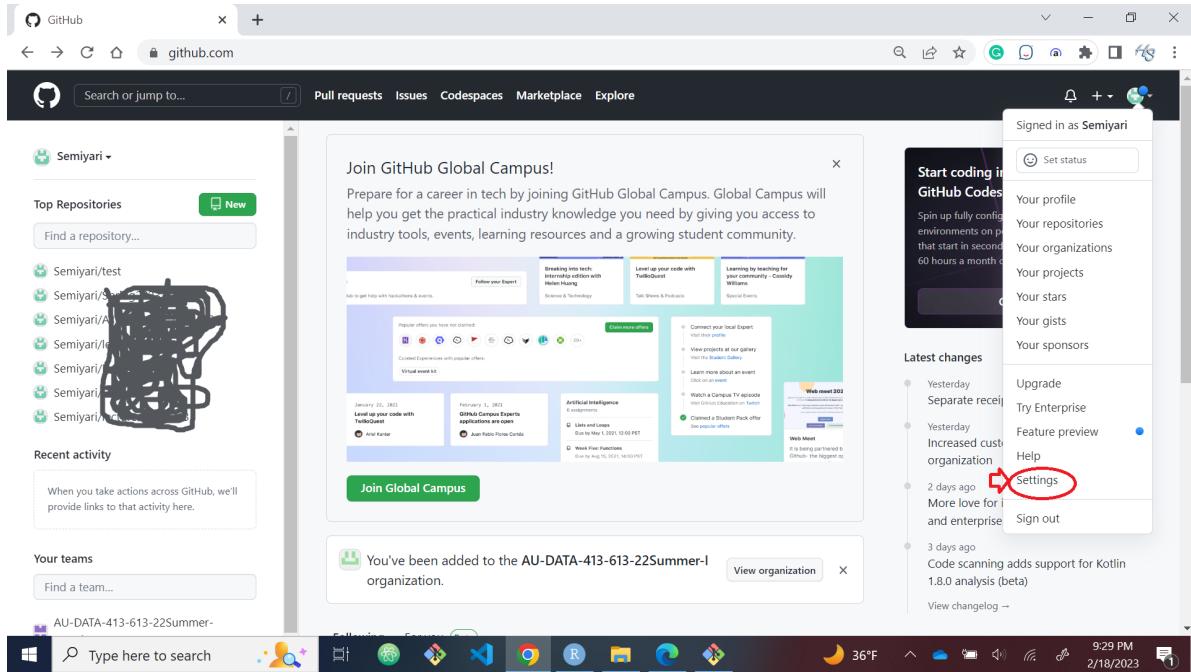


- Step 4. You can retrieve your recovery code (for Two-Factor Authentication) here

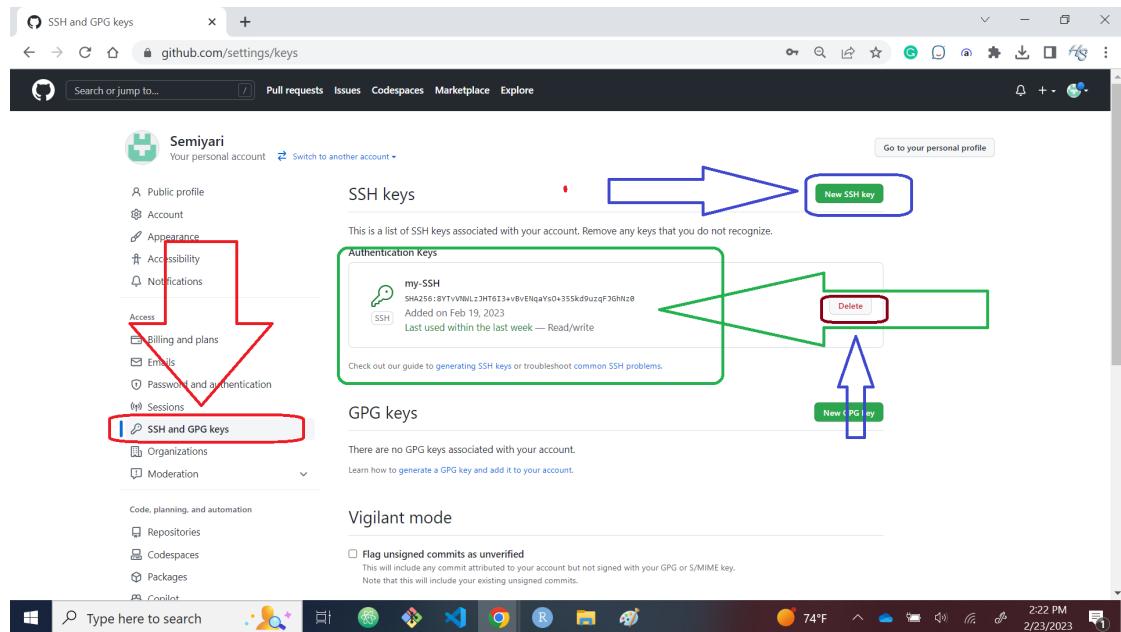


## SSH

- Step 1. In the upper-right corner of any page, click on your avatar (or your profile photo), then click *Settings*.



- Step 2. Scroll down and click on “SSH and GPG”. If you have an SSH Key you will see the following. It also gives you option to add more keys or Delete the current one,



## PAT

- Step 1. In the upper-right corner of any page, click on your avatar (or your profile photo), then click *Settings*.

Signed in as Semiyari

Start coding in GitHub Codes

Your profile  
Your repositories  
Your organizations  
Your projects  
Your stars  
Your gists  
Your sponsors

Latest changes

- Yesterday Separate receipt
- Yesterday Increased cust organization
- 2 days ago More love for I and enterprise
- 3 days ago Code scanning adds support for Kotlin 1.8.0 analysis (beta)

Settings

9:29 PM 2/18/2023

Join GitHub Global Campus!

Prepare for a career in tech by joining GitHub Global Campus. Global Campus will help you get the practical industry knowledge you need by giving you access to industry tools, events, learning resources and a growing student community.

Follow your Expert  
Level up your code with Twinkl  
Learning by teaching for Helen Huang  
Science & Technology  
Talk Shows & Podcasts

Join Global Campus

You've been added to the AU-DATA-413-613-22Summer-I organization.

Your Profile

github.com/settings/profile

Pages  
Saved replies  
Security  
Code security and analysis  
Integrations  
Applications  
Scheduled reminders  
Archives  
Security log  
Sponsorship log  
Developer settings

Company

Location

Display current local time

Update profile

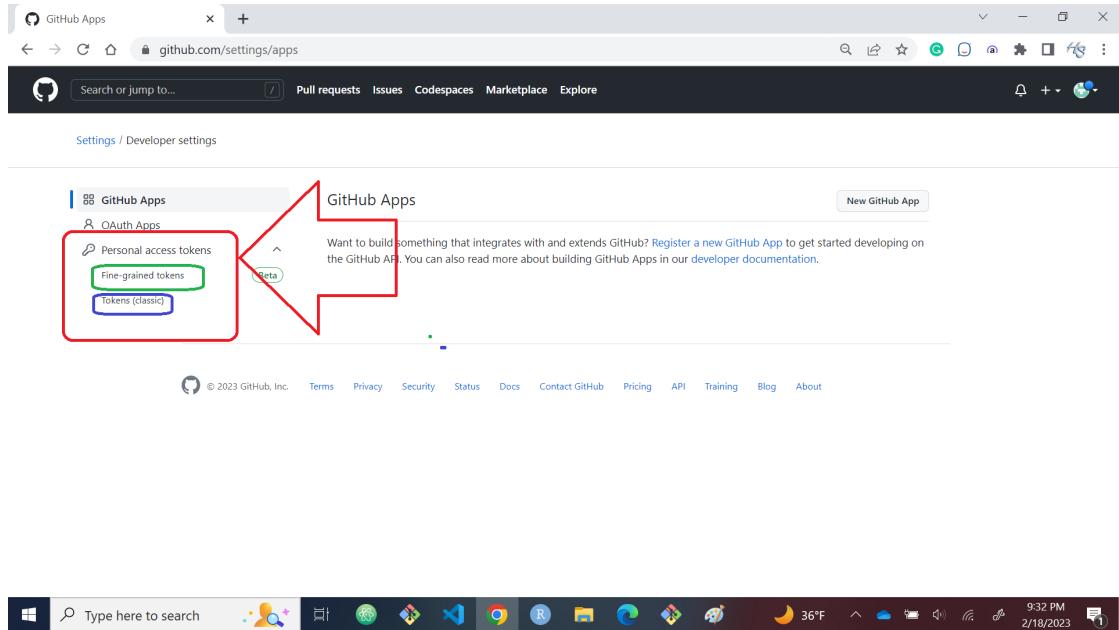
Contributions & Activity

Make profile private and hide activity  
Include private contributions on my profile

Update preferences

9:31 PM 2/18/2023

- Step 2.
- Step 3. Select “Token (classic)”



- Step 4. You will see your Token here. If you do not have any token then you must generate one. See the note “authorization” to refresh your memory

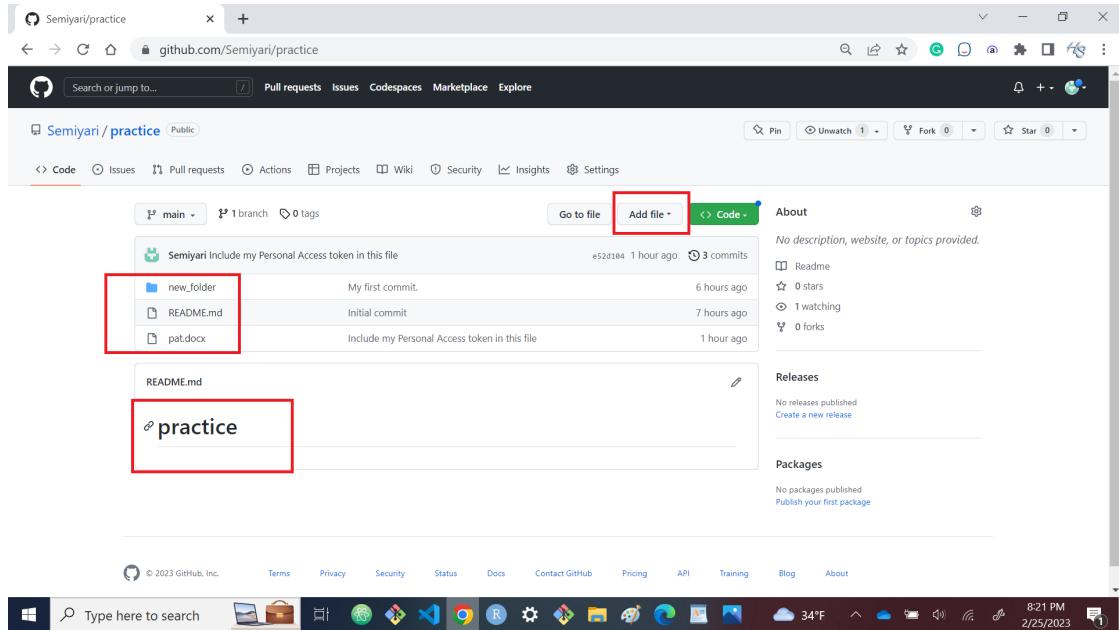
```
! [](./01_figs/pat3cls1.png)
```

## Adding/Pushing a file to a repository on GitHub

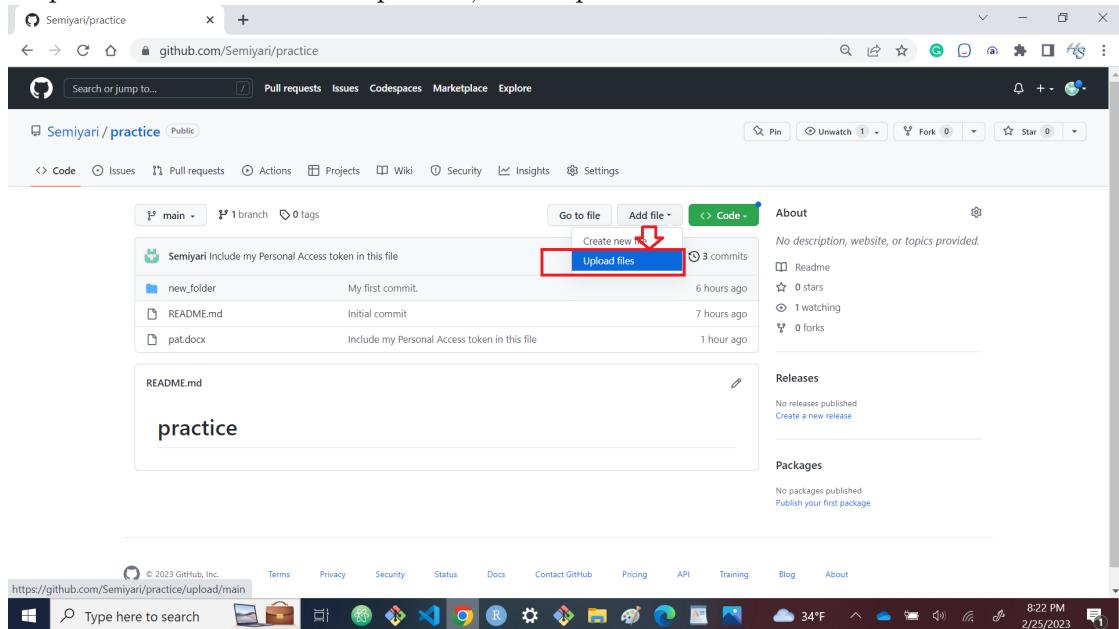
- Files that you add to a repository via a browser are limited to 25 MB per file. You can add larger files, up to 100 MB each, via the command line. For more information, see “[Adding a file to a repository using the command line.](#)” To add files larger than 100 MB, you must use Git Large File Storage. For more information, see “[About large files on GitHub.](#)”

### Add to a repository via a browser

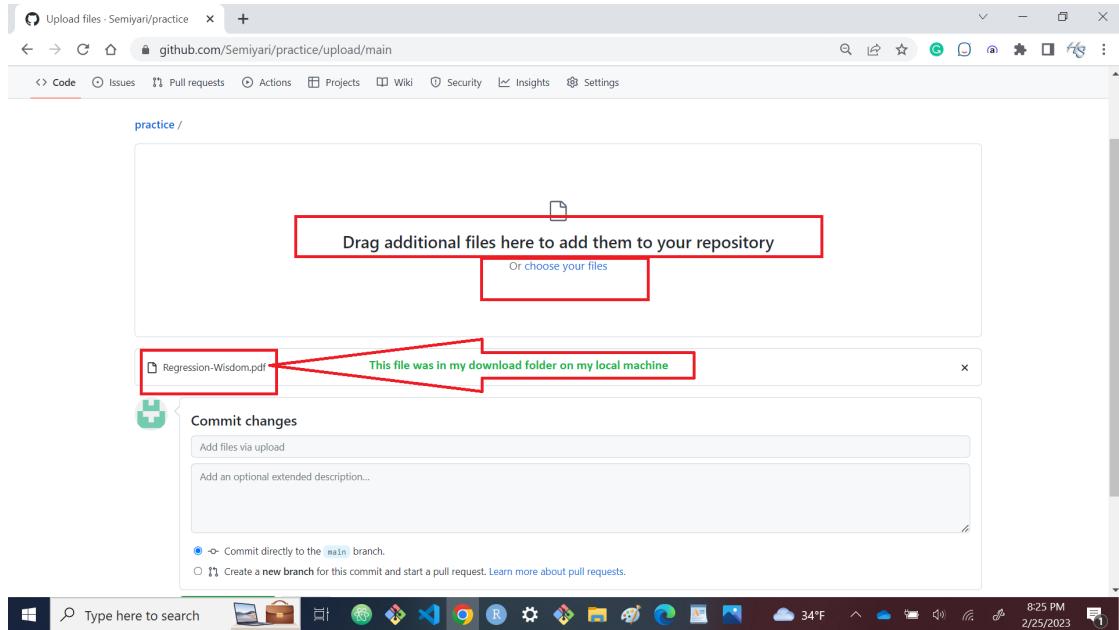
- Files that you add to a repository via a browser are limited to 25 MB per file.
- Step 1. On GitHub.com, navigate to the main page of the repository. Above the list of files, using the Add file drop-down, click Upload files.



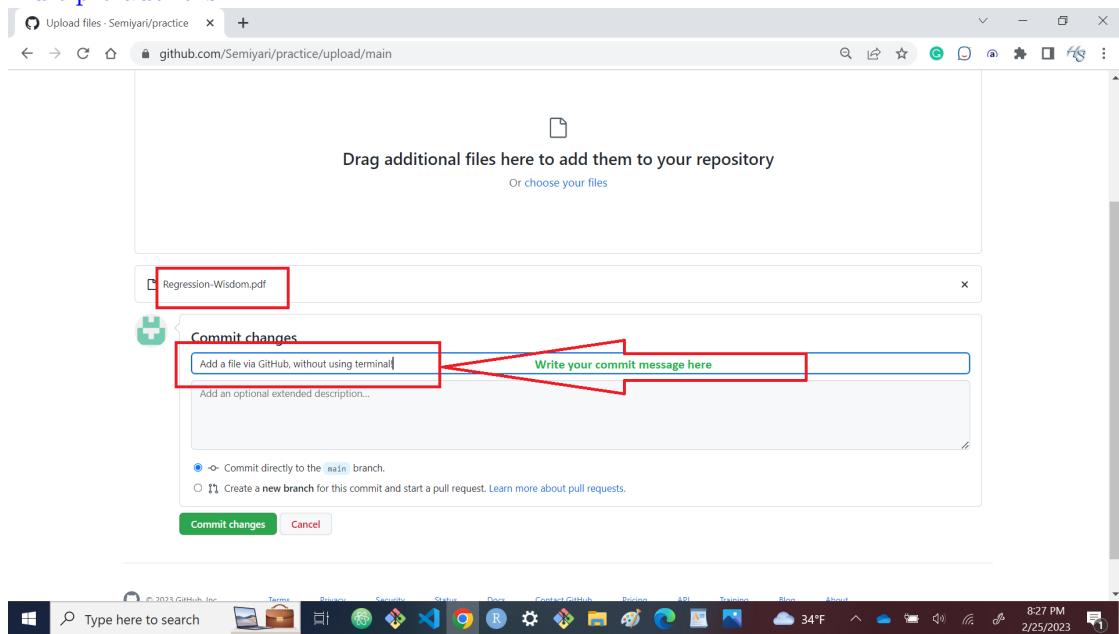
- Step 2. Click on Add file drop-down, click Upload files.



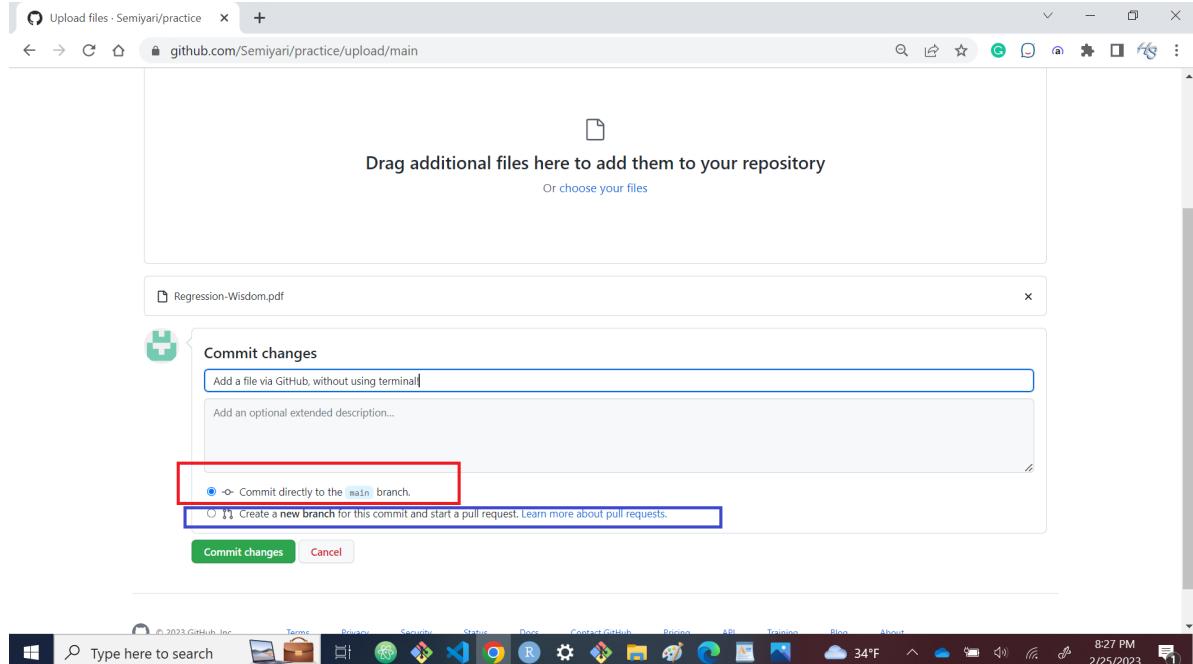
- Step 3. Drag and drop the file or folder you'd like to upload to your repository onto the file tree.



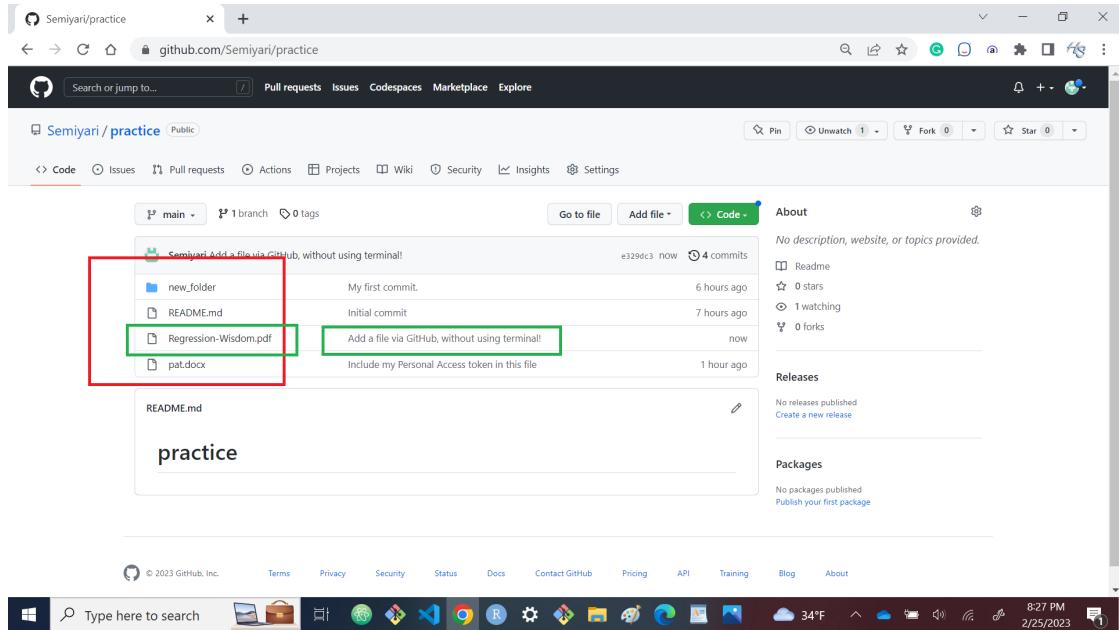
- Step 4. At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in the commit message. For more information, see “[Creating a commit with multiple authors](#).”



- Step 5. Below the commit message fields, decide whether to add your commit to the current branch or to a new branch. If your current branch is the default branch, you should choose to create a new branch for your commit and then create a pull request. For more information, see “[Creating a pull request](#).”



- Step 6. Click Commit changes.

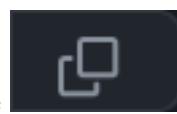


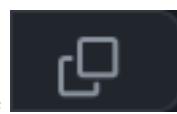
## Adding a file to a repository using the command line

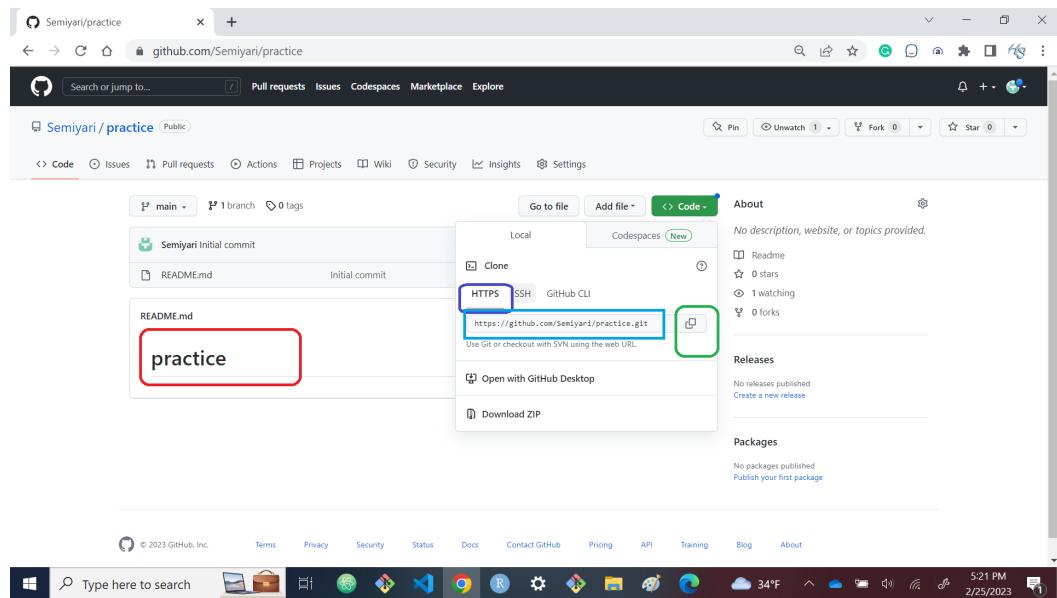
- Files that you add to a repository via a command line are limited to 100 MB per file.

## Pushing a file with Two-Factor:

- Step 1. Creating a New Repository
  - Repository name is “practice”.
  - The repo is set to be “Public”
  - You check “Add a README File”
- Step 2. Cloning:
  - using “HTTPS”
- Step 3. Make sure that “HTTPS” is highlighted.



– Then click on the  button to copy the link.



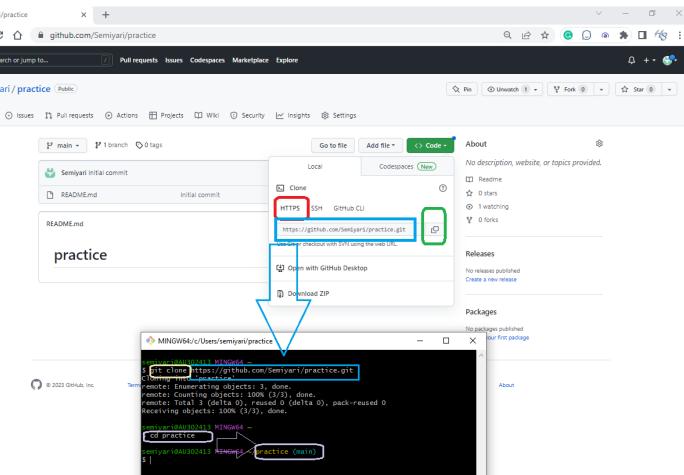
- Step 4. Open terminal and type

```
git clone <URL>
```

- Step 5. Change the directory

```
cd practice
```

- Check local machine to see if your repository is there.



- Step 6. Make a directory call it assignment

```
mkdir new_folder
```

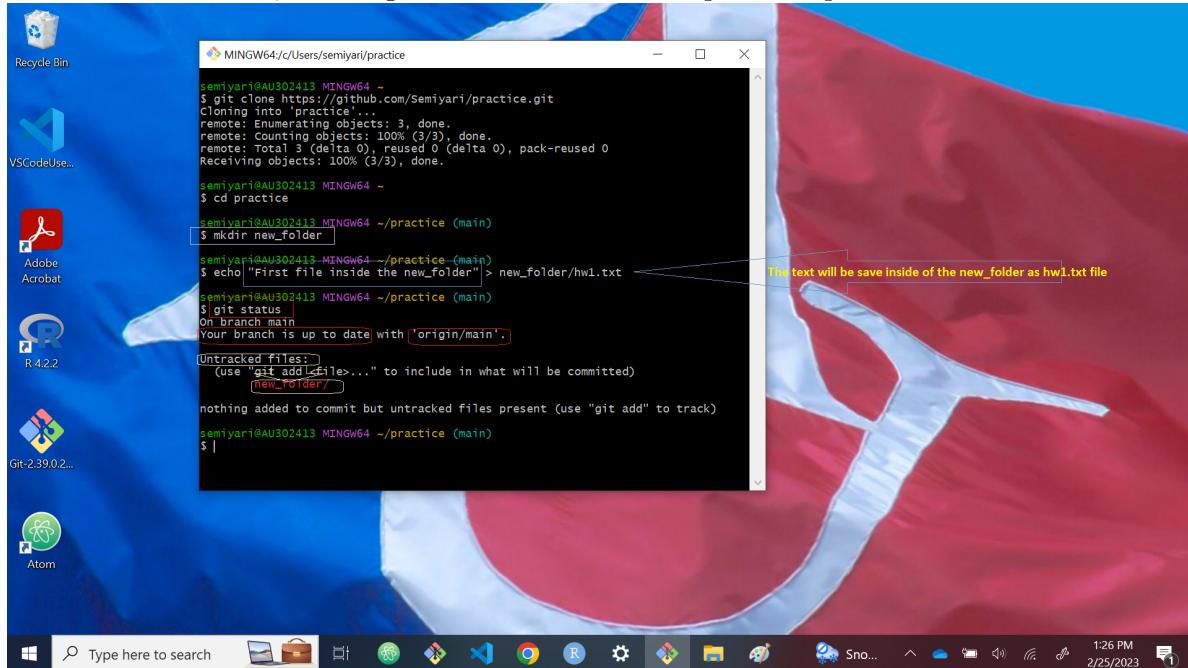
- Step 7. Make first file and place it into the new\_folder in your repository **practice** (not really a homework assignment)

```
echo "First file inside the new_folder" > new_folder/hw1.txt
```

- get status of your local machine

```
git status
```

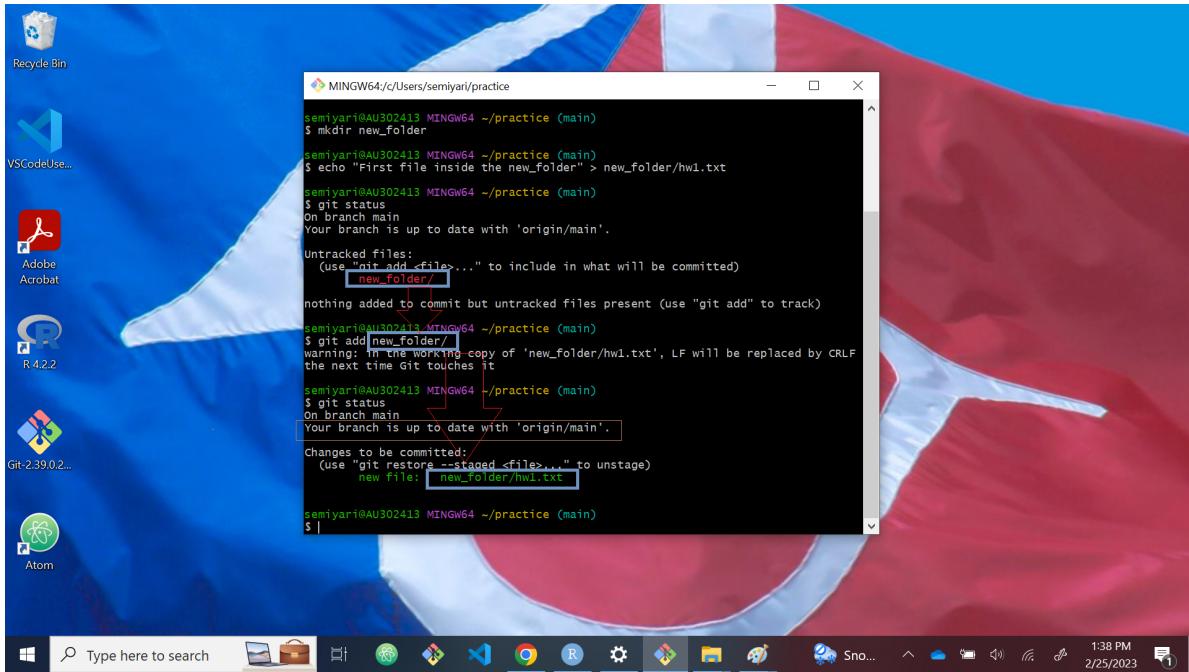
-The file is untracked, meaning that Git sees a file not part of a previous commit.



- Step 8. Move changes from the working directory to the Git staging area, by `git add` command. The staging area is where you prepare a snapshot of a set of changes before committing them to the official history.

```
git add new_folder/
```

```
git status
```



- Step 9. Commit the file. Here we had a change on a file in one of the folder in our repository. Use `git commit` command. Don't forget your message

```
git commit -m "My first commit."
```

A screenshot of a Windows desktop environment. On the left, there's a vertical column of icons for various applications: Recycle Bin, VSCodeUse..., Adobe Acrobat, R 4.2.2, Git-2.39.0.2..., and Atom. The main window is a terminal session titled 'MINGW64/c/Users/semiyari/practice'. The terminal shows the following command-line session:

```

warning: in the working copy of 'new_folder/hw1.txt', LF will be replaced by CRLF
the next time Git touches it

semiyari@AU302413 MINGW64 ~/practice (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new_folder/hw1.txt

semiyari@AU302413 MINGW64 ~/practice (main)
$ git commit -m "My first commit."
[main d00f053] My first Commit
 1 file changed, 1 insertion(+)
 create mode 100644 new_folder/hw1.txt

semiyari@AU302413 MINGW64 ~/practice (main)
$ git log
commit a944161449eaea886d81652d9d558f4f53cf2d33 (HEAD -> main)
Author: Hamid Semiyari <semyari1@american.edu>
Date:   Sat Feb 25 13:41:09 2023 -0500

  Initial commit

semiyari@AU302413 MINGW64 ~/practice (main)
$ |

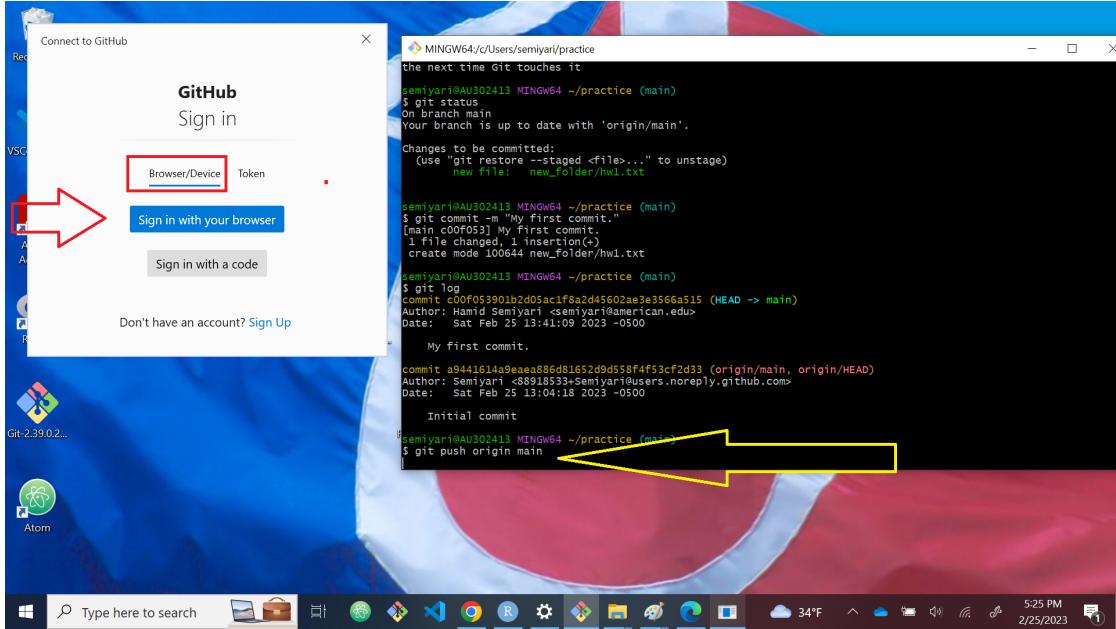
```

The terminal window has a dark theme. The commit message 'My first commit.' is highlighted with a yellow rectangular selection.

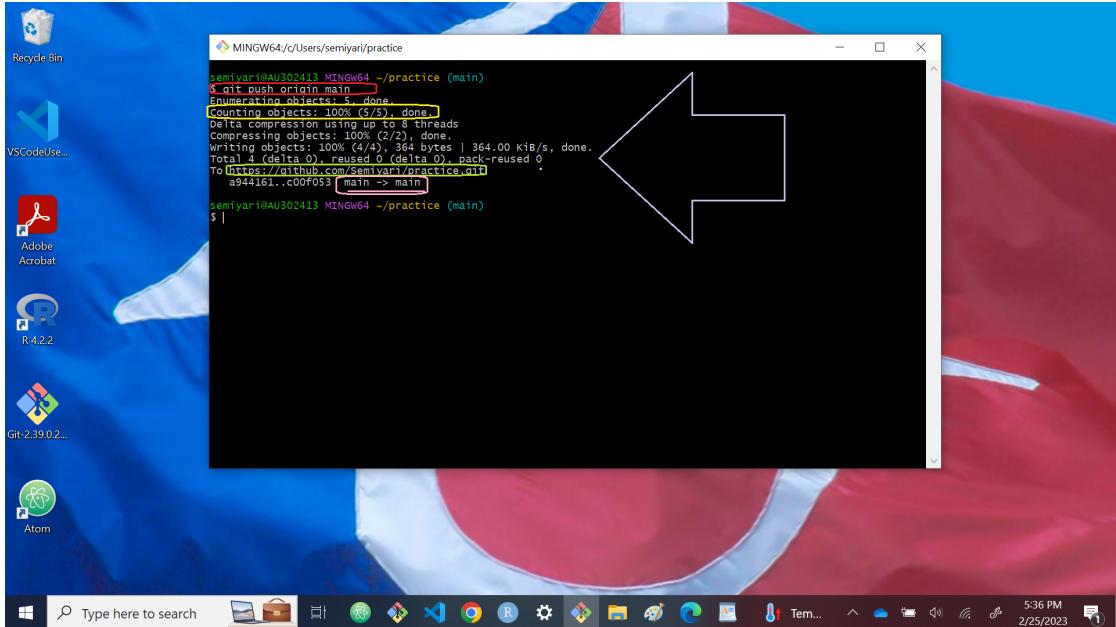
- Step 10. Use the `git push` commands to push the changes to the GitHub repository. `origin` parameter tells Git to push the changes you made in your local (your laptop) repository to remote (your GitHub) repository. Git also pushes the changes on your local `main` branch to the `main` branch of your main repository.

`git push origin main`

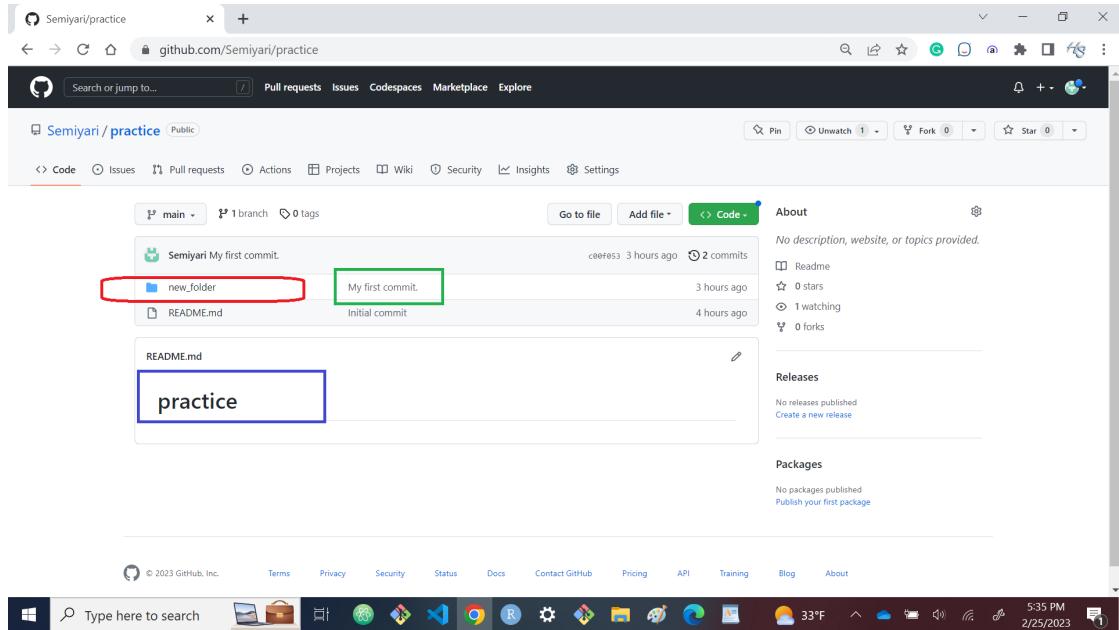
- The signing page will be open and ask to sign in. Select “Browser/Device”.



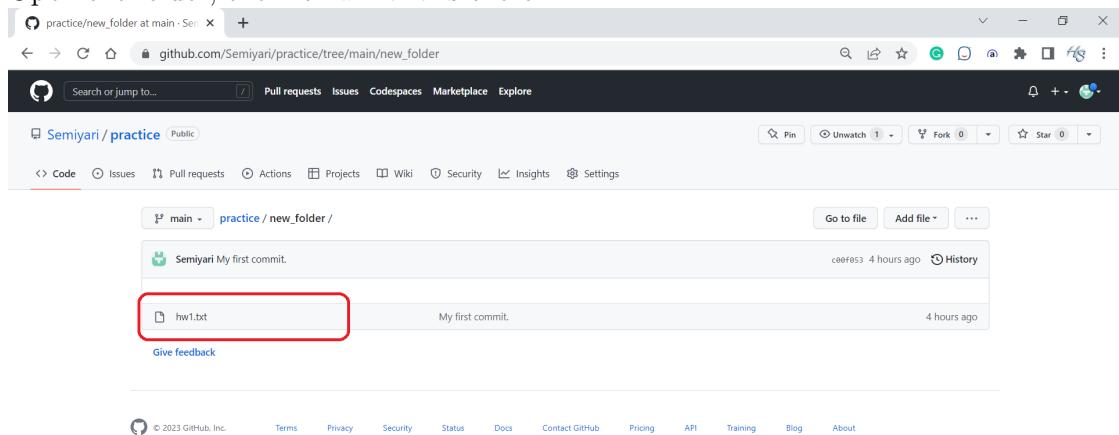
- Check your terminal



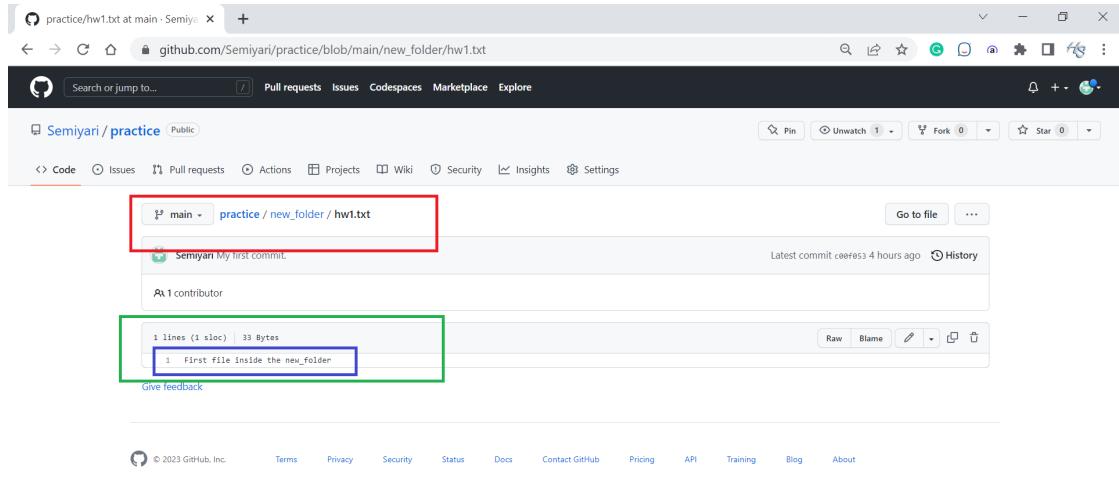
- Check your GitHub the new folder, `new_folder`, is added



- Open the folder, the file **hw1.txt** is there

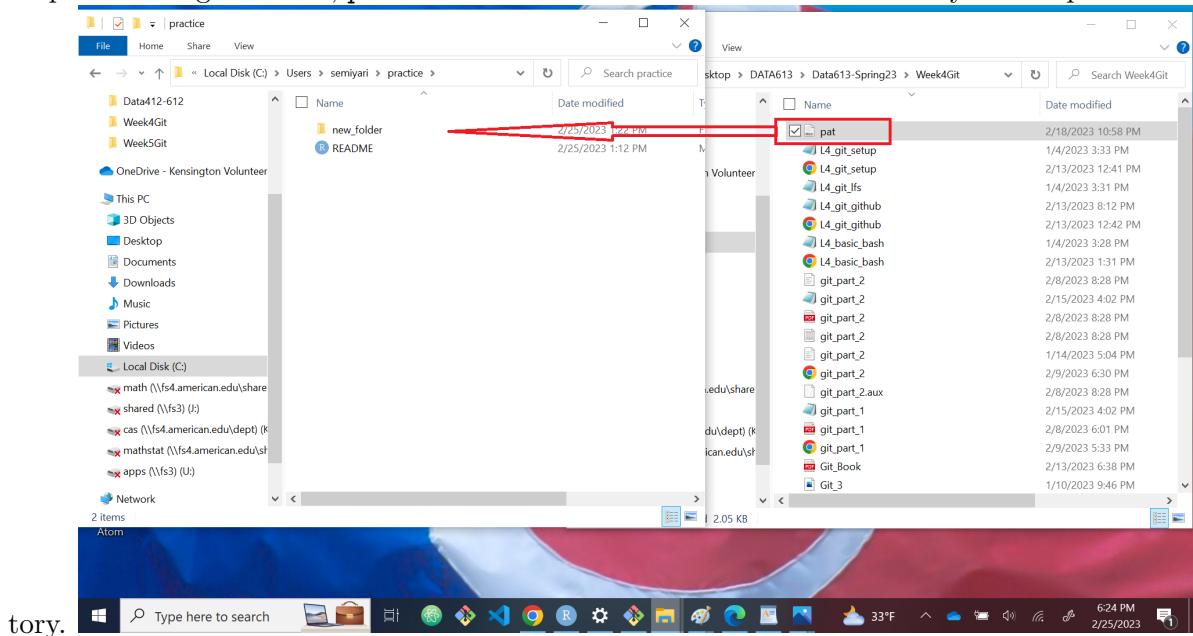


- Open the file **hw1.txt**. You will see the first line that we just wrote in.

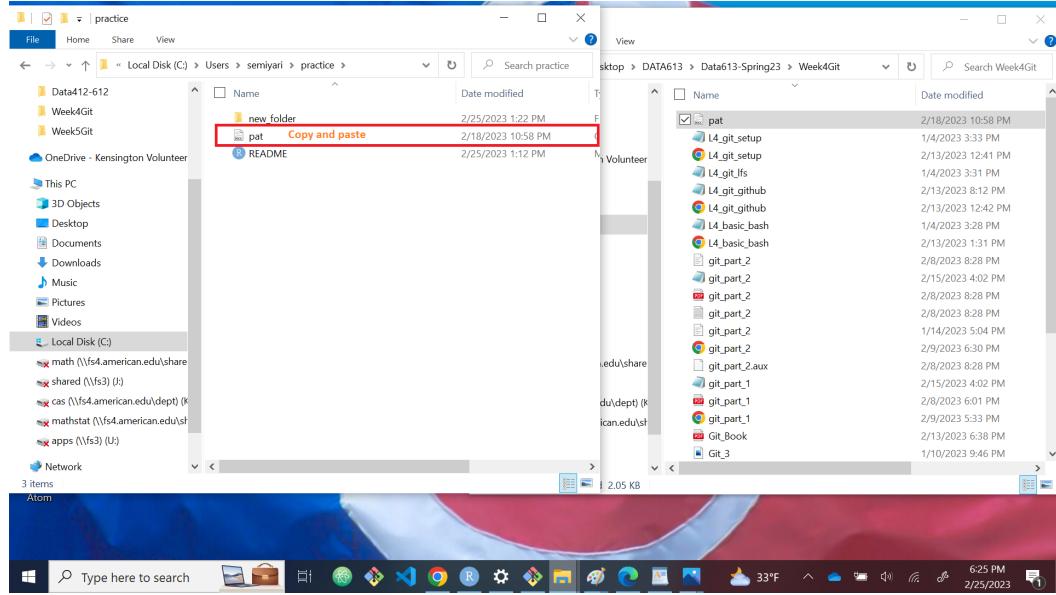


## Pushing by Personal Access Token

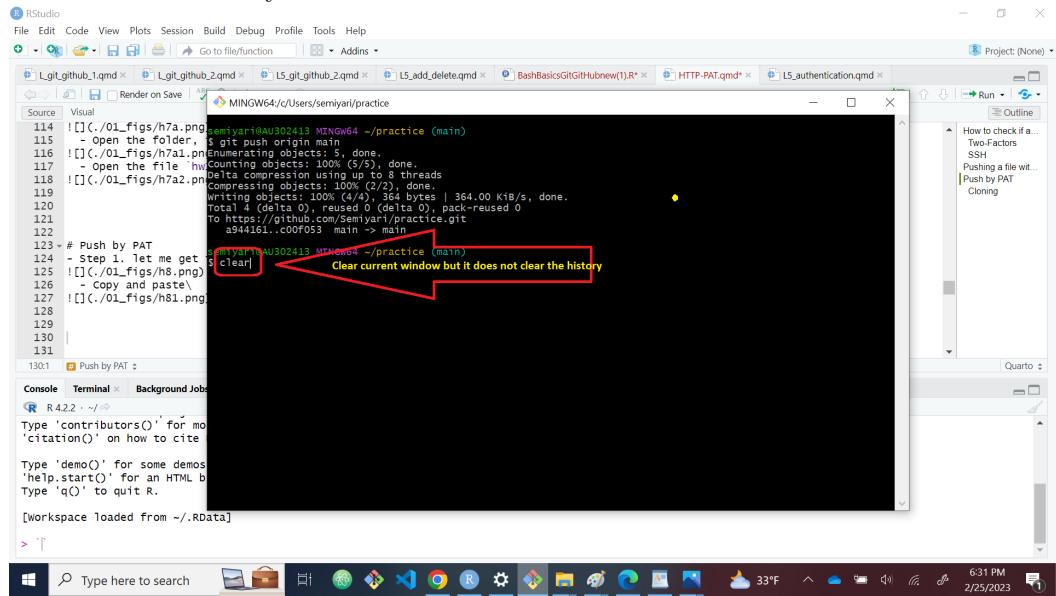
- Step 1. let me get the file, pat.docx from another folder and move it to my local repository.



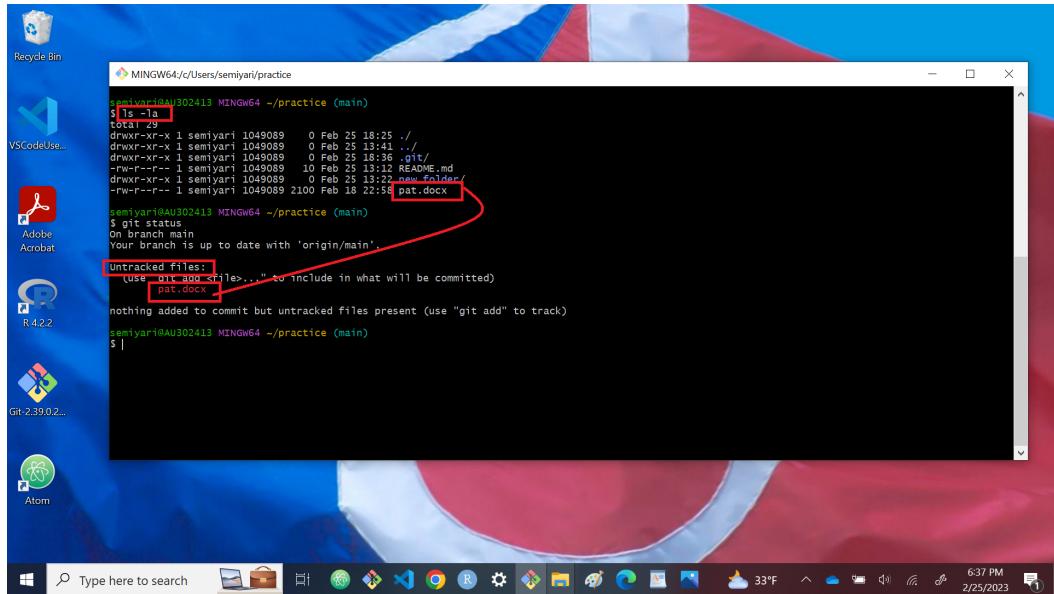
– Copy and paste



– clean the terminal by command `clear`. The short cut is `CTRL+L`



– Type command `ls -la` to see the content in your local repository.



- Check the status
- Step 2. Move changes by `git add` command to staging area.

```
git add pat.docx
```

```
git status
```

- Commit the file

```

MINGW64:/c/Users/semyari/practice
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>" to include in what will be committed)
    pat.docx

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

semmyari@AU302413 MINGW64 ~/practice (main)
$ git add pat.docx
fatal: pathspec 'pat.docx' did not match any files

semmyari@AU302413 MINGW64 ~/practice (main)
$ git add pat.docx
semmyari@AU302413 MINGW64 ~/practice (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>" to unstage)
    new file:   pat.docx

semmyari@AU302413 MINGW64 ~/practice (main)
$ git commit -m "Include my Personal Access token in this file"
[main e6e6e6] include my Personal Access token in this file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 pat.docx

semmyari@AU302413 MINGW64 ~/practice (main)
$ git remote set-url origin https://semmyari:ghp_dvjkZ79Em7zb9zay2jyEAE1MfV1M2nf5g

```

- You need to copy your PAT

If you come to this page and do not see any token then you need to Generate a new token

This is my PAT

Copy The PAT



- Step 3: We need to set the remote repository URL by running the command

```
git remote set-url origin https://<USER>:<PAT>@github.com/<USER>/<REPOSITORY>.git
```

For example mine is:

```
git remote set-url origin https://semiyari:ghp_dwjKZ79Em7z0b9zay2jyEAE1MmFVlM2nf5g4@github.com
```

- Then you need to type

```
git push origin main
```

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with various icons. Below it is a browser window titled "Semiyari/practice" showing the GitHub repository page for "Semiyari/practice". In the center, a terminal window is open with the following command history:

```
git remote set-url origin https://semiyari:ghp_dwjKZ79Em7z0b9zay2jyEAE1MmFVlM2nf5g4@github.com
git push origin main
```

The terminal shows the command being typed, followed by the output of the command. The output includes:

- \$ git status  
On branch main  
Your branch is up to date with 'origin/main'.
- Changes to be committed:  
(use "git restore --staged <file>..." to unstage)  
new file: pat.docx
- git commit -m "Include my Personal Access token in this file"  
[main e52d104] Include my Personal Access token in this file  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 10644 pat.docx
- git remote set-url origin https://semiyari:ghp\_dwjKZ79Em7z0b9zay2jyEAE1MmFVlM2nf5g4@github.com
- git push origin main

The terminal prompt shows the command being completed: \$ git push origin main

A purple box highlights the URL in the terminal command: `https://semiyari:ghp_dwjKZ79Em7z0b9zay2jyEAE1MmFVlM2nf5g4@github.com`. Another purple box highlights the word "PAT" in the commit message. A third purple box highlights the word "Repository" in the GitHub browser page.

- Step 4. The last step is to check your GitHub to see if your file is there

