

Push

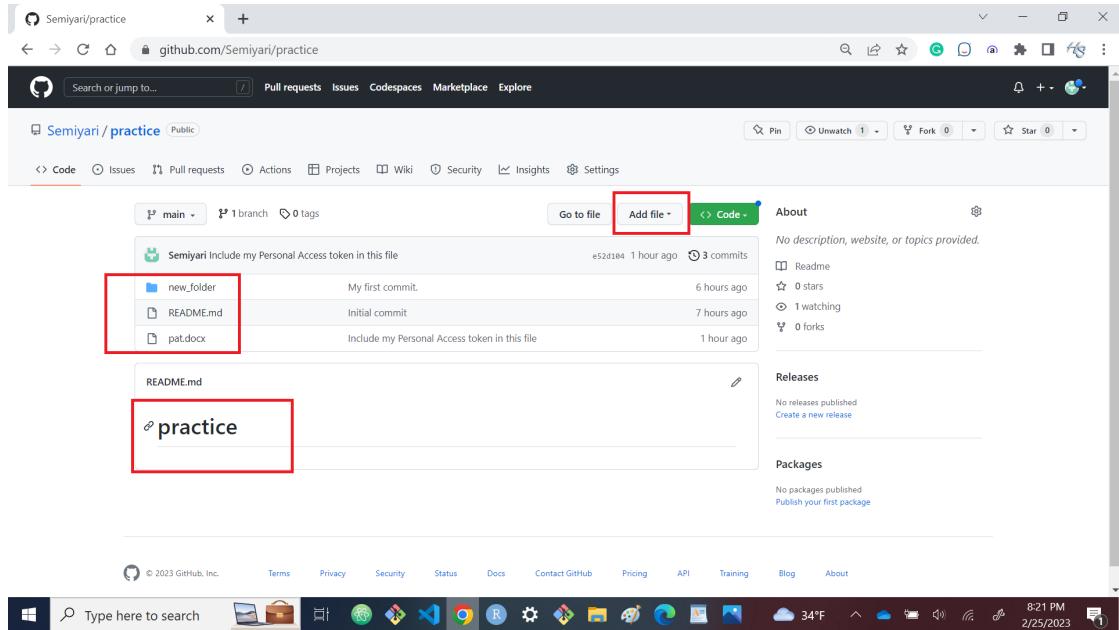
Hamid Semiyari

Adding/Pushing a file to a repository on GitHub

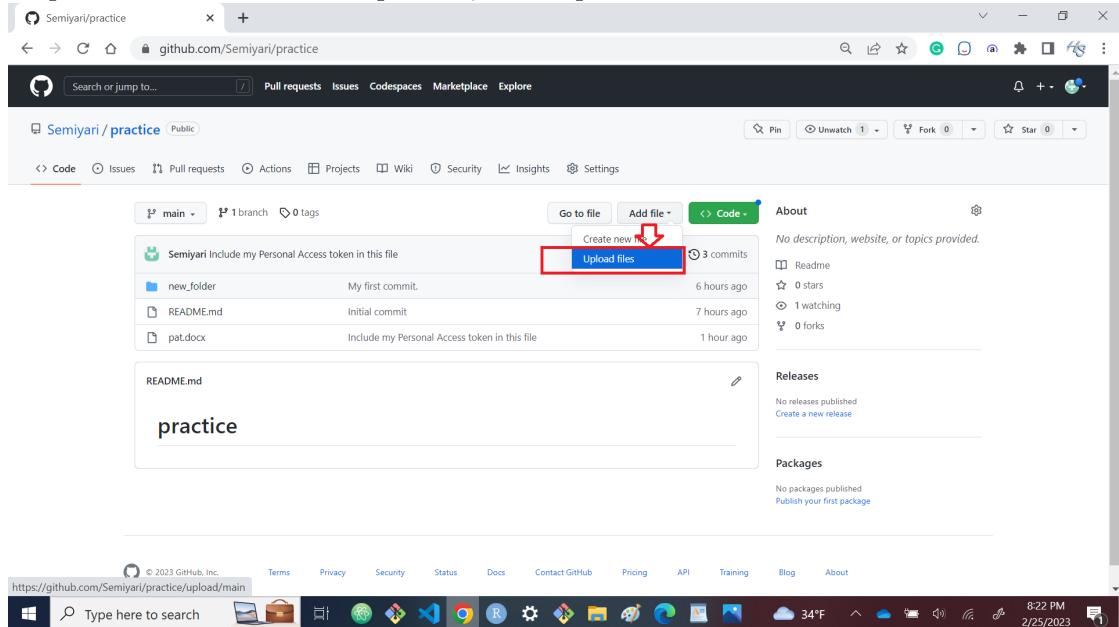
- Files that you add to a repository via a browser are limited to 25 MB per file. You can add larger files, up to 100 MB each, via the command line. For more information, see “[Adding a file to a repository using the command line.](#)” To add files larger than 100 MB, you must use Git Large File Storage.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

Add to a repository via a browser

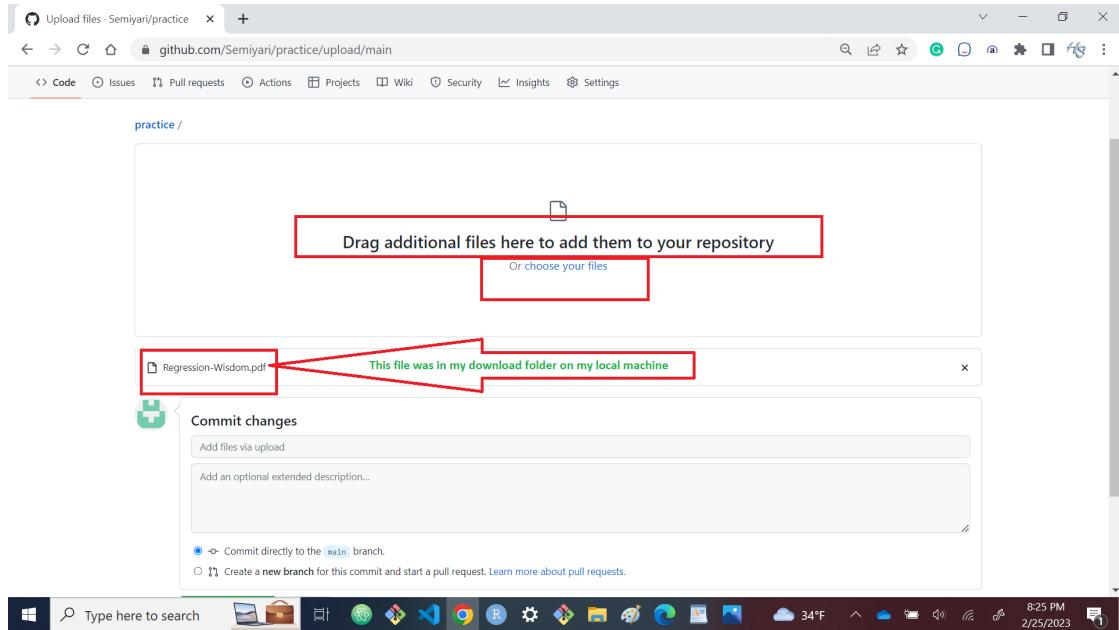
- Files that you add to a repository via a browser are limited to 25 MB per file.
- Step 1. On GitHub.com, navigate to the main page of the repository. Above the list of files, using the Add file drop-down, click Upload files.



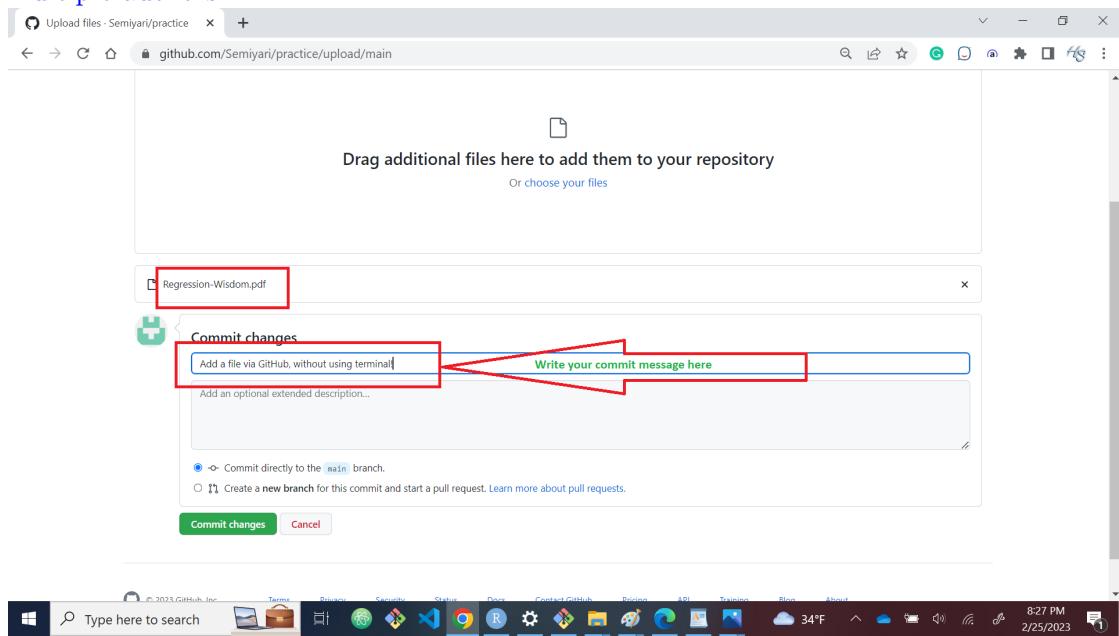
- Step 2. Click on Add file drop-down, click Upload files.



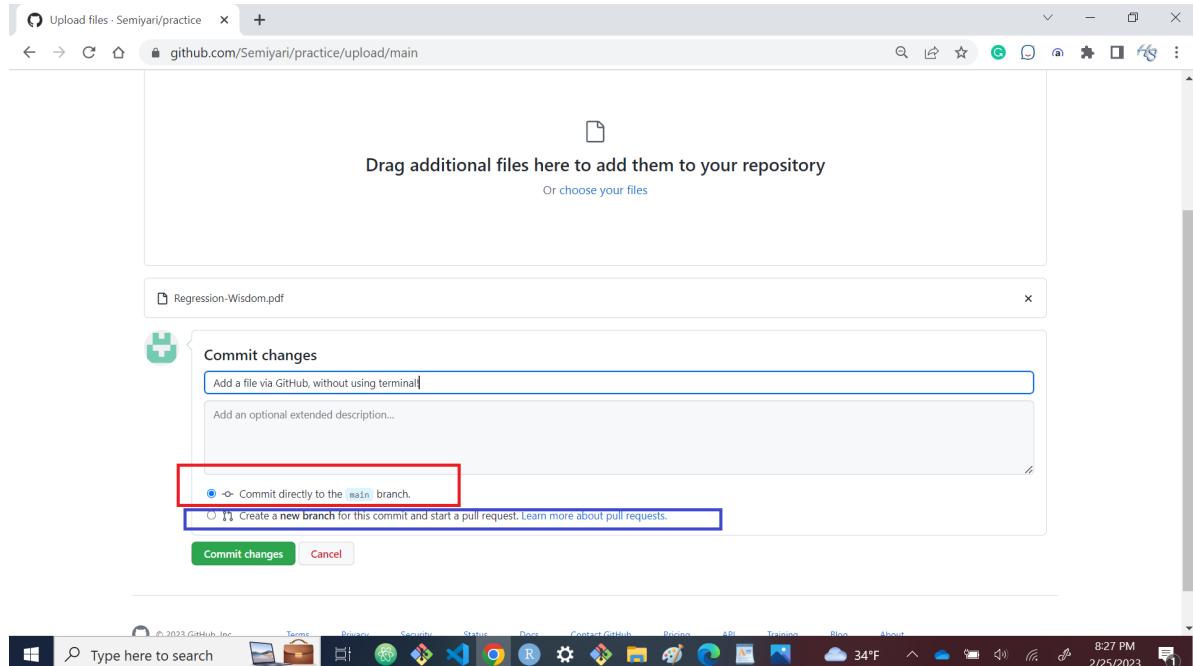
- Step 3. Drag and drop the file or folder you'd like to upload to your repository onto the file tree.



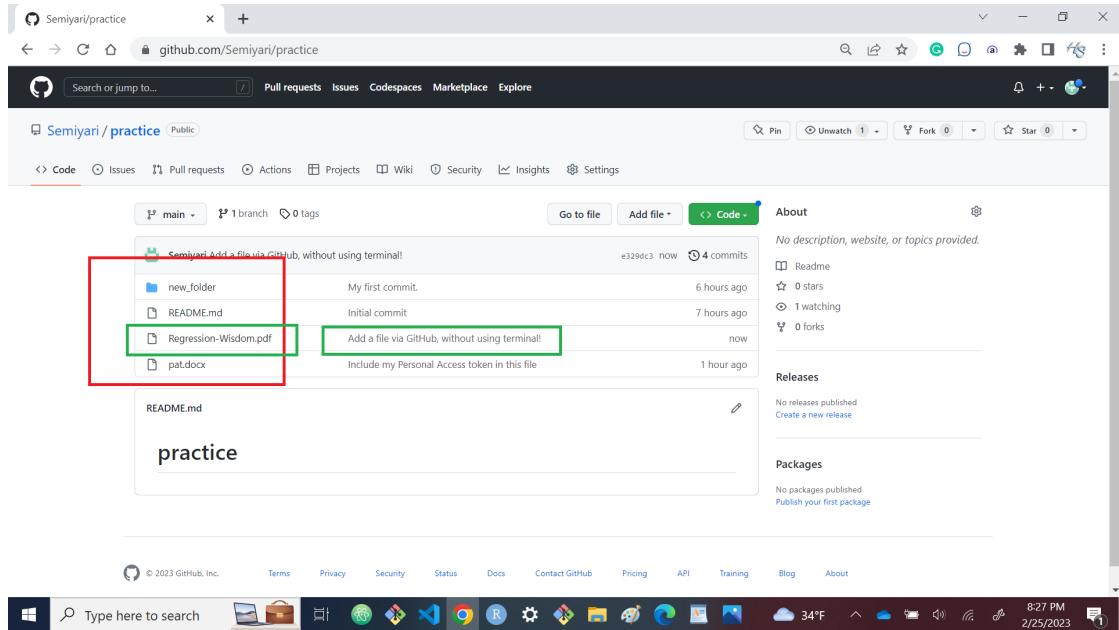
- Step 4. At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in the commit message. For more information, see “[Creating a commit with multiple authors](#).”



- 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](#).”



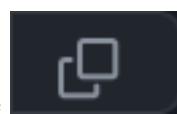
- Step 6. Click Commit changes.

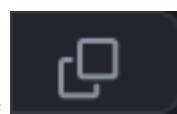


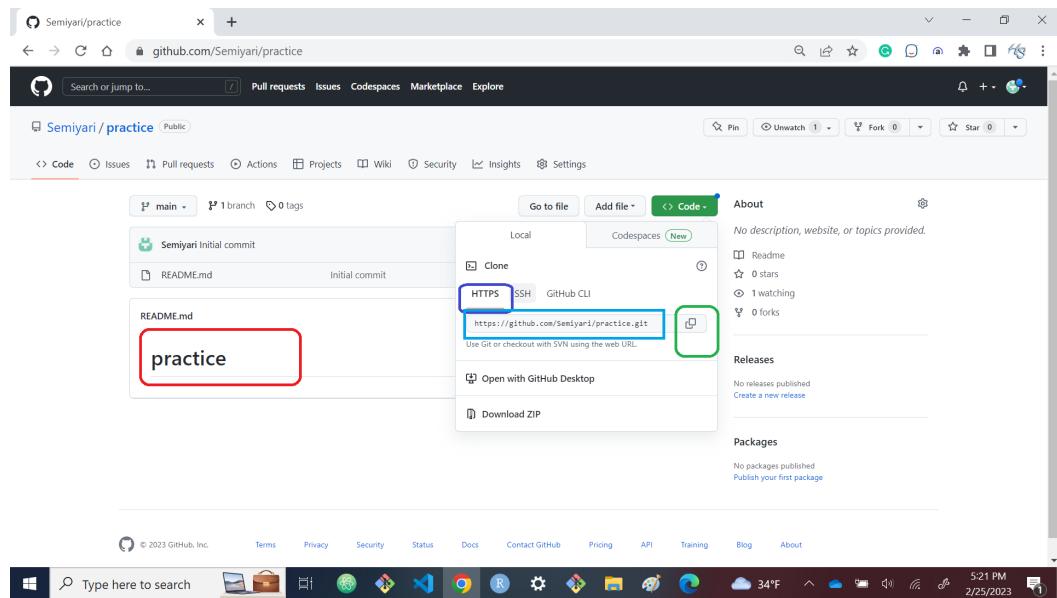
Adding a file to a repository using the command line

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.



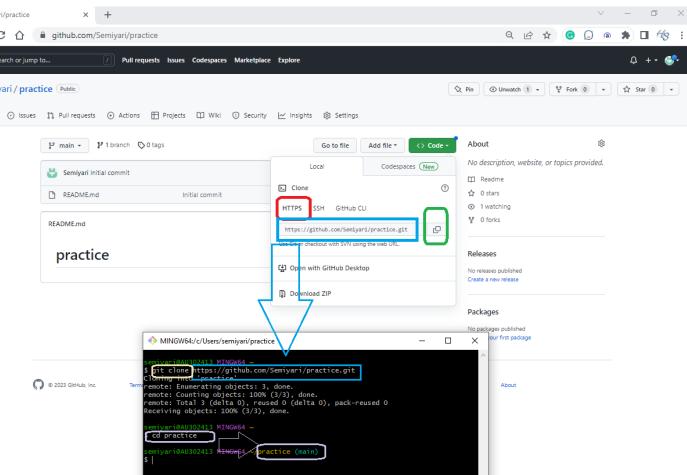
- Step 4. Open terminal and type

```
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
```

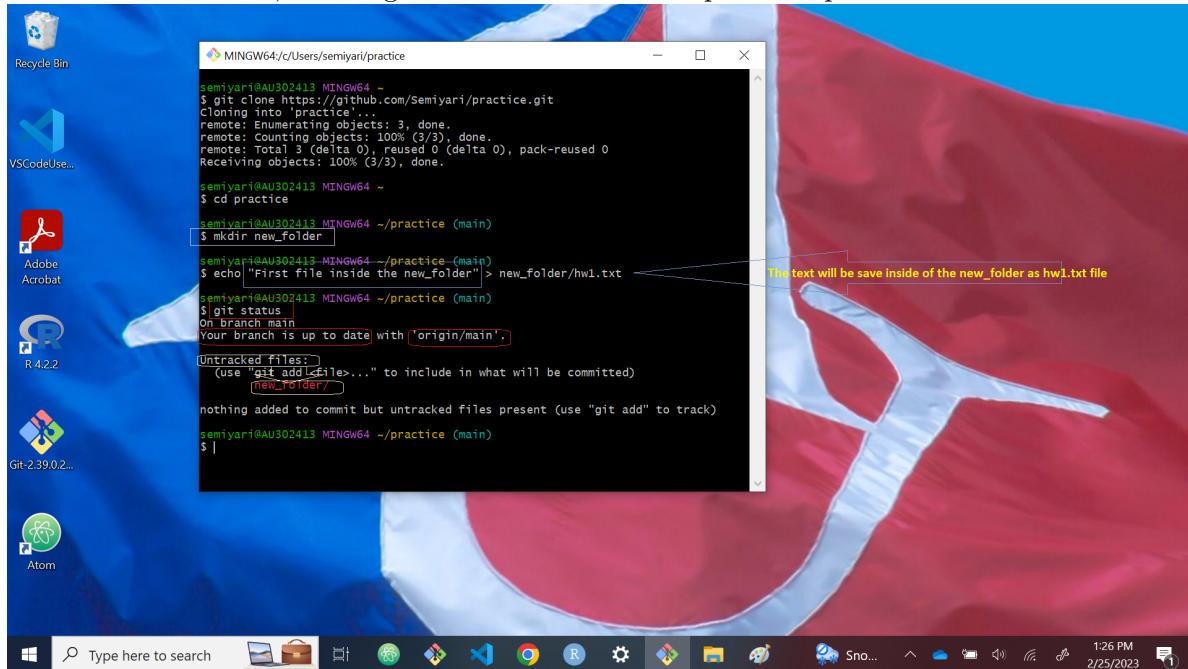
- Step 7. Make first file and place it into the new_folder in your repository **practice** (not really a homework assignment)

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

- get status of your local machine

```
git status
```

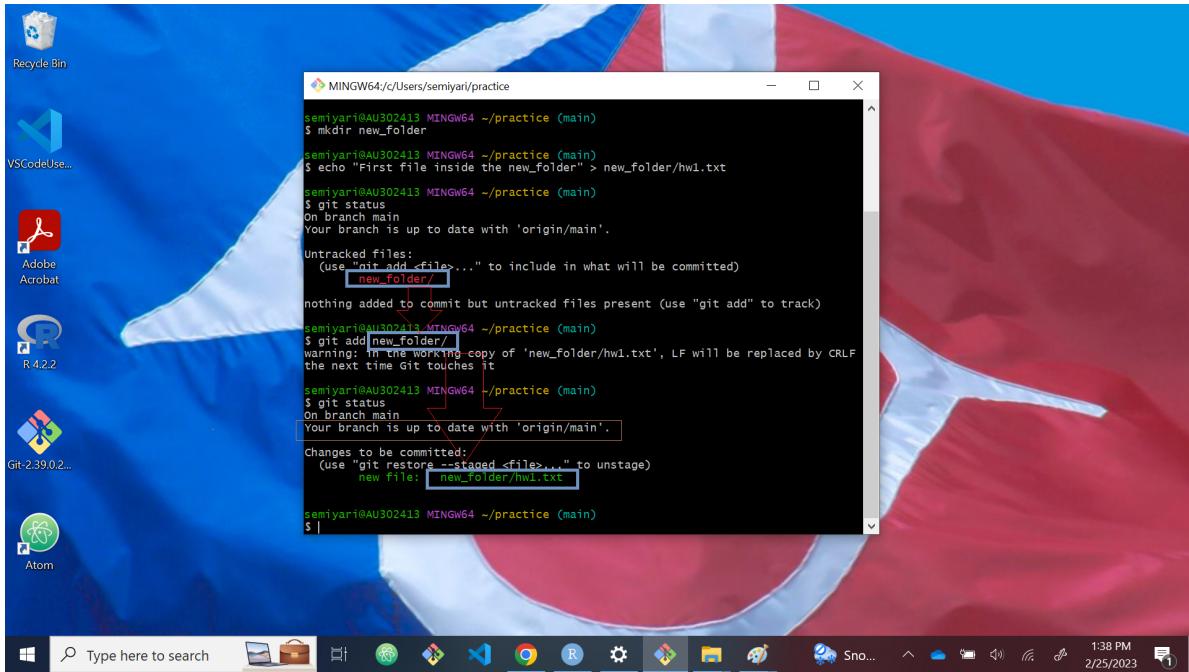
The file is untracked, meaning that Git sees a file not part of a previous commit.



- 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
```



A screenshot of a Windows desktop environment. The background features the Chilean flag. On the left, there is 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 taskbar at the bottom contains several pinned icons: Microsoft Edge, File Explorer, Task View, Start button, Taskbar settings, Snipping Tool, and a system tray with a battery icon, signal strength, and the date/time (1:38 PM, 2/25/2023). A search bar on the taskbar says "Type here to search". In the center, a terminal window titled "MINGW64/c/Users/semiyari/practice" is open, showing a sequence of git commands. Red arrows point from the text in the terminal to specific parts of the command line, highlighting the creation of a new folder, the addition of files to the staging area, and the commit message.

```
semiyari@AU302413 MINGW64 ~/practice (main)
$ mkdir new_folder
semiyari@AU302413 MINGW64 ~/practice (main)
$ echo "First file inside the new_folder" > new_folder/hw1.txt
semiyari@AU302413 MINGW64 ~/practice (main)
$ git status
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)
    [new_folder]

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

semiyari@AU302413 MINGW64 ~/practice (main)
$ git add [new_folder]
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)
$ |
```

- Step 9. Commit the file. Here we had a change on a file in one of the folder in or 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)
$ |

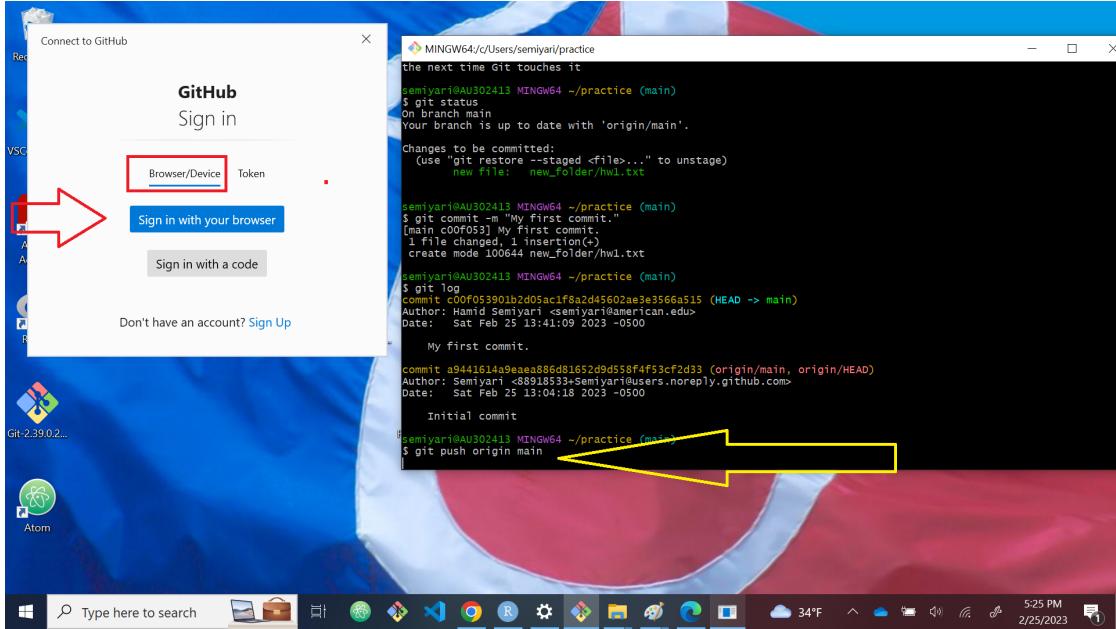
```

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

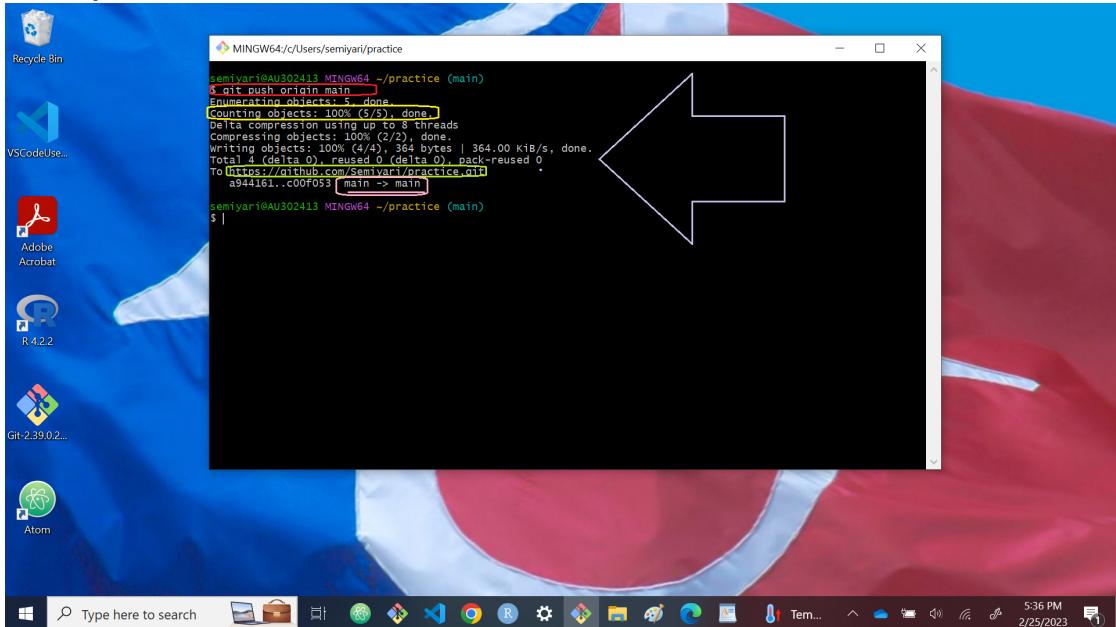
- Step 10. Use the `git push` commands to push the changes to the GitHub repository. `origin` parameter tells Git to push the changes you made in your local (your laptop) repository to remote (your GitHub) repository. Git also pushes the changes on your local `main` branch to the `main` branch of your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

`git push origin main`

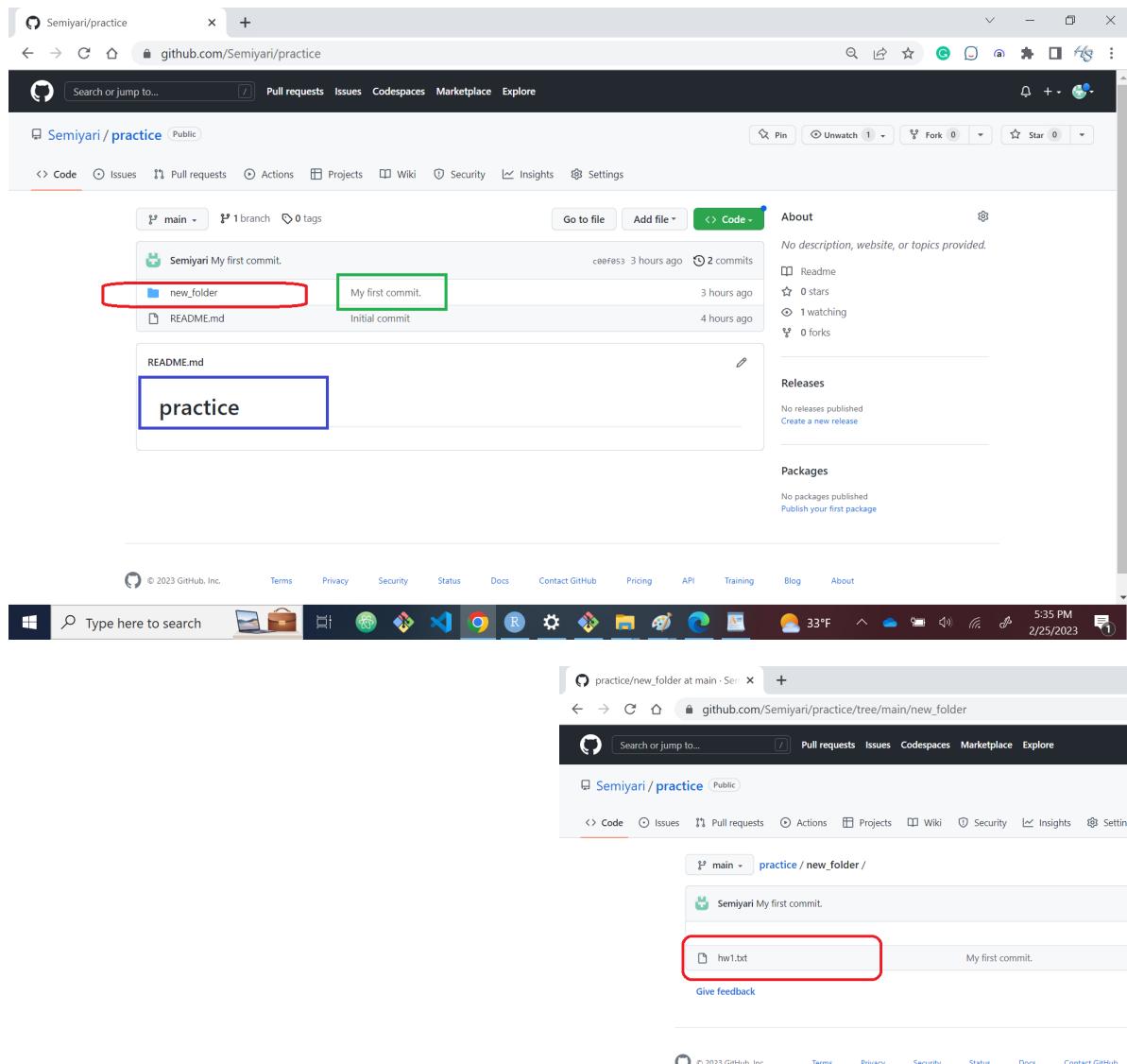
- The signing page will be open and ask to sign in. Select “Browser/Device”.



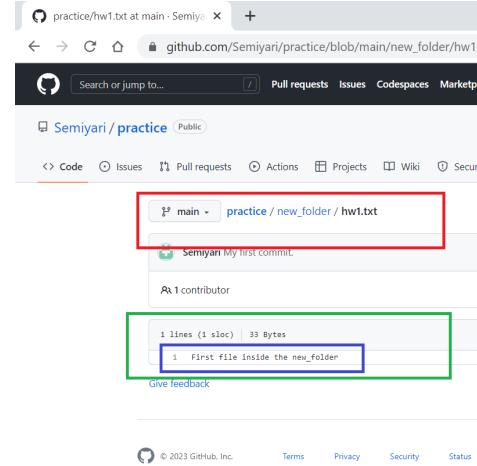
- Check your terminal



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



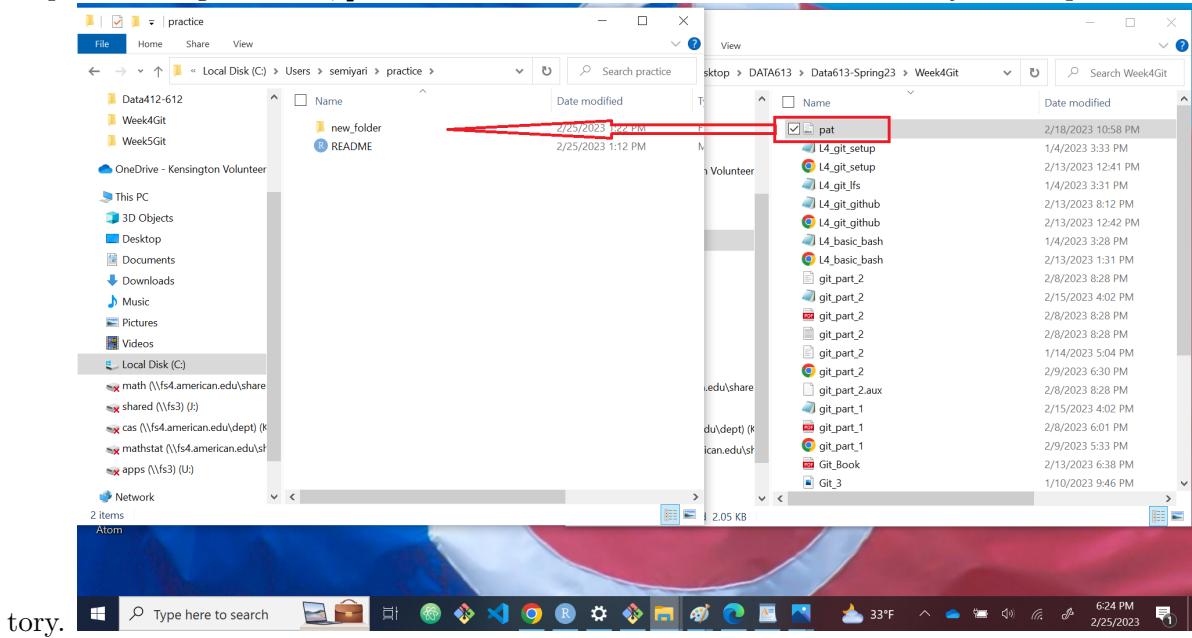
- Open the folder, the file `hw1.txt` is there



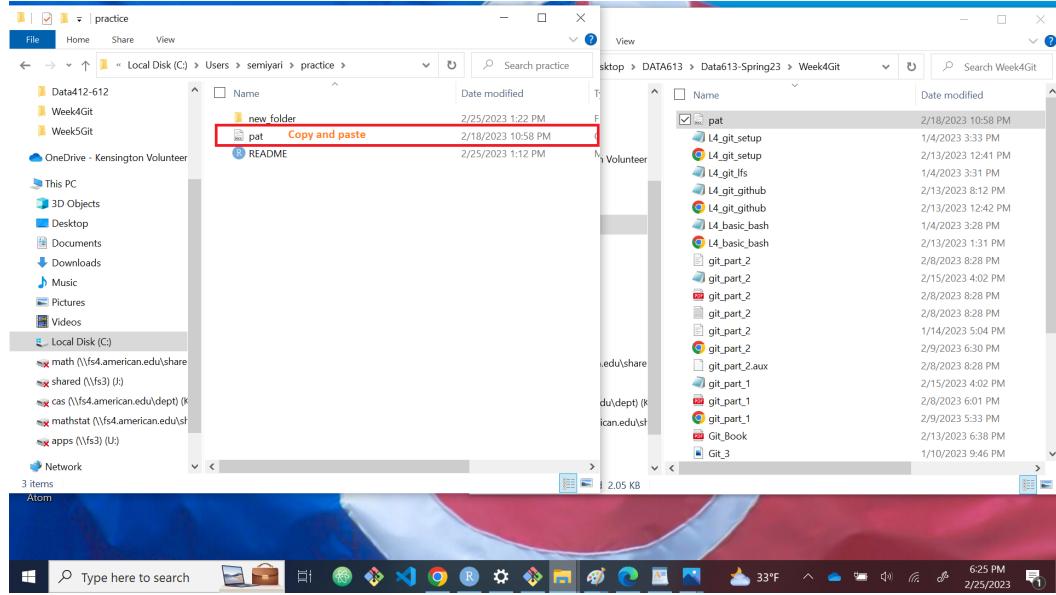
- Open the file `hw1.txt`. You will see the first line that we just wrote in.

Pushing by Personal Access Token

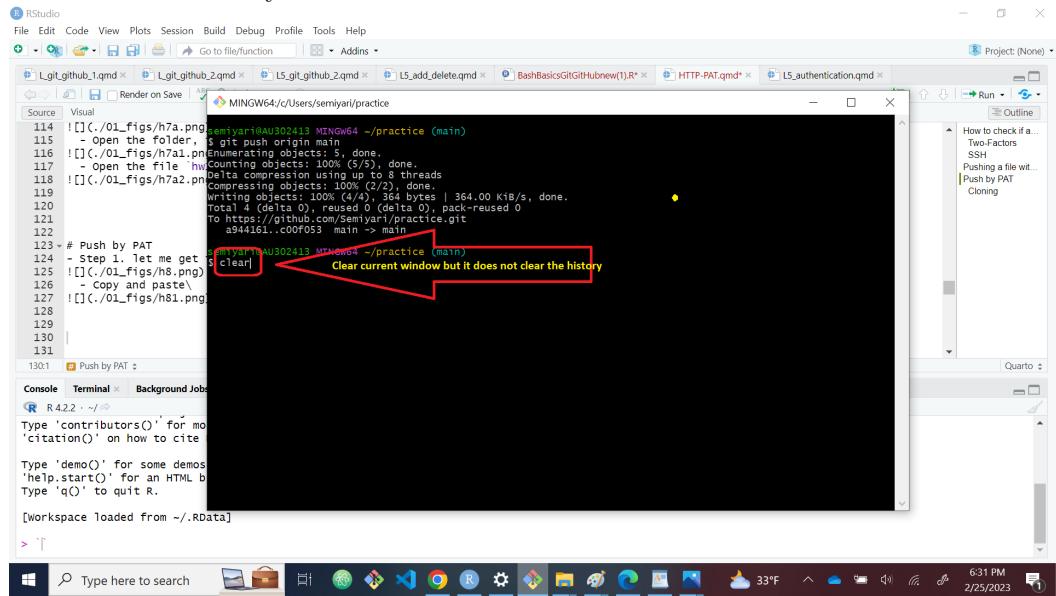
- Step 1. let me get the file, `pat.docx` from another folder and move it to my local repository.



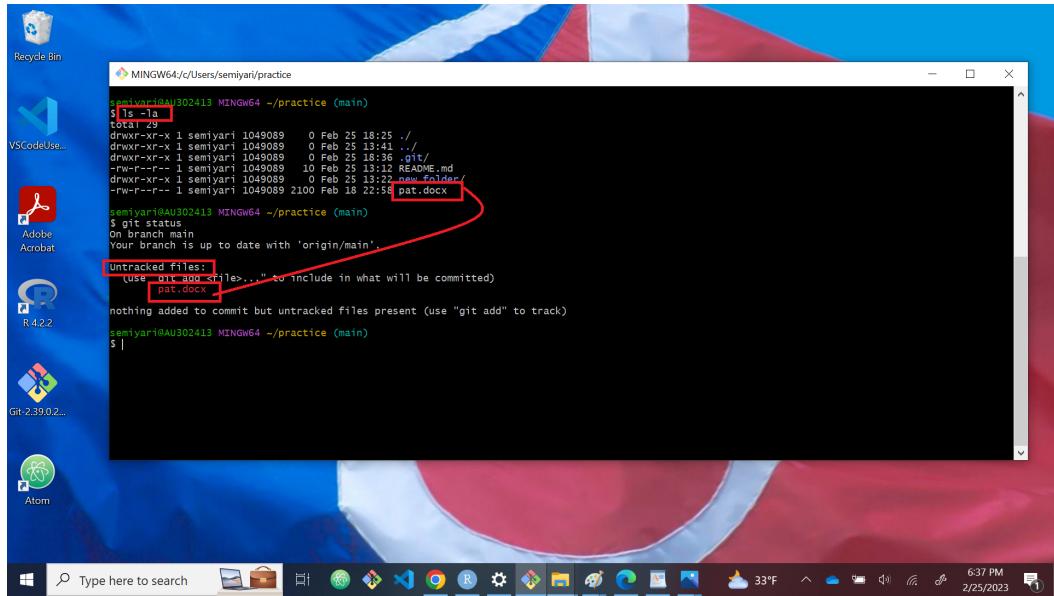
– Copy and paste



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



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



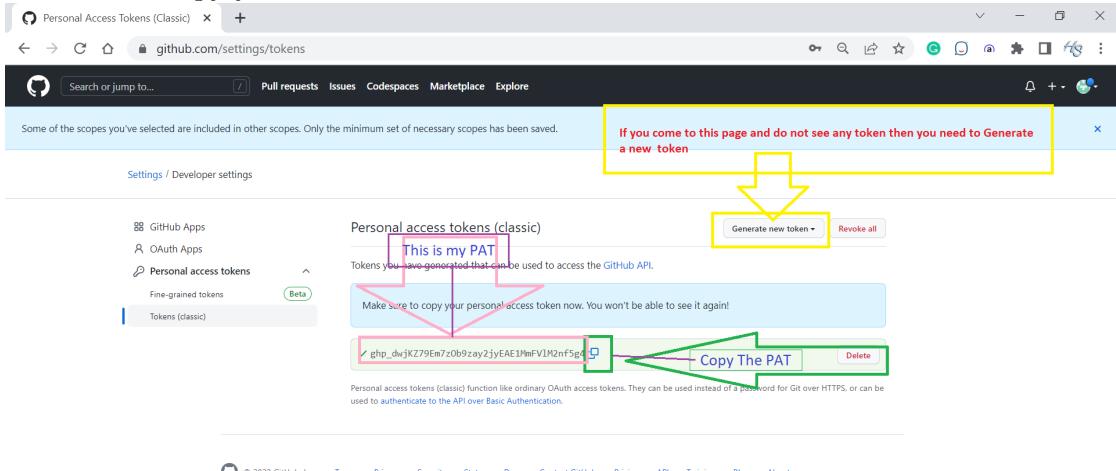
- Check the status
- Step 2. Move changes by `git add` command to staging area.

```
git add pat.docx
```

```
git status
```

- Commit the file

- You need to copy your PAT



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

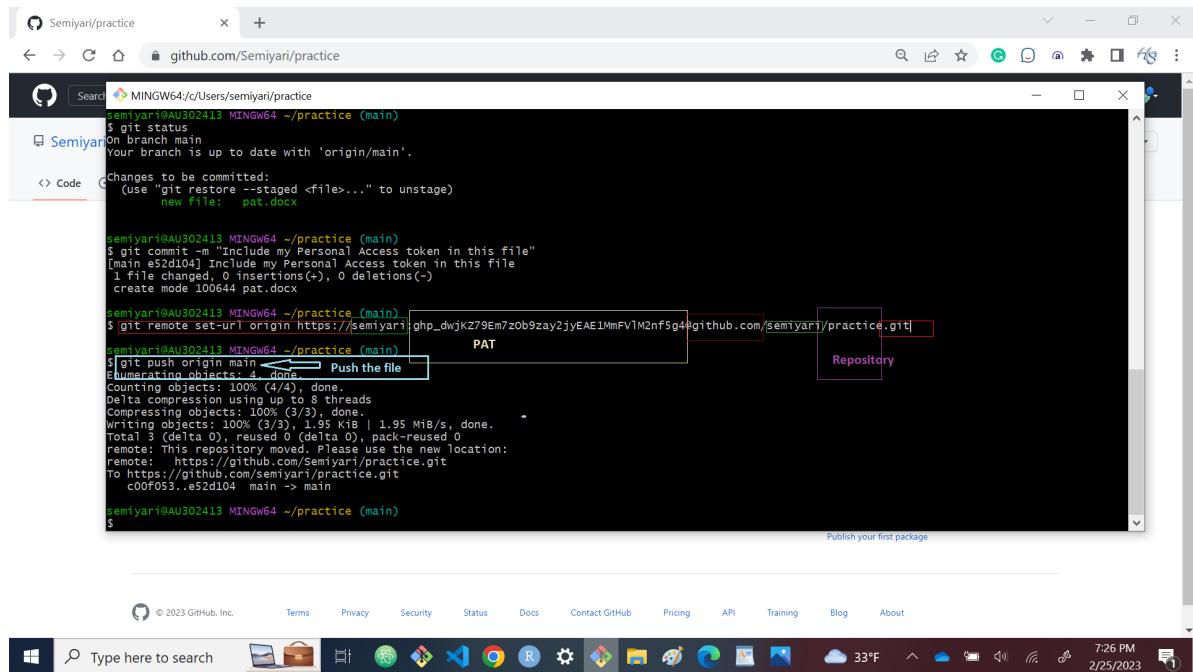
```
git remote set-url origin https://<USER>:<PAT>@github.com/<USER>/<REPOSITORY>.git
```

For example mine is:

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

- Then you need to type

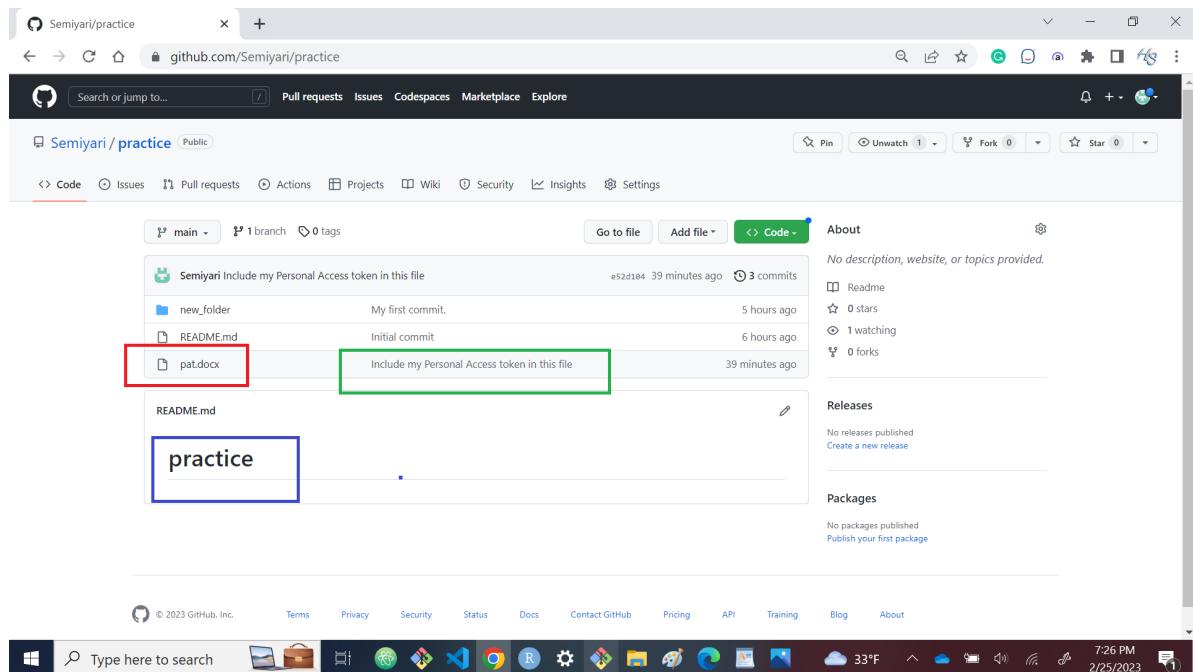
```
git push origin main
```



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

```
git push origin main
```

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



- Step 10. At your GitHub repository's page, click to copy the remote repository URL.
- Step 11. Enter: git remote add

```
git remote add <unique_nickname> URL(without https://)@PAT
#(this will set the new remote). It is better not to use a nickname twice!
```

- Step 12. Enter:

```
git remote -v
# (this will verify the new remote URL)
```

- Step 13 Enter:

```
git push -u <nickname> master
# (this code will push the file to Github)
```

- Step 14. Go to Github and locate your file

Cloning

- using “HTTPS”

- Make sure that “HTTPS” is highlighted.



- Then click on the button to copy the link.

