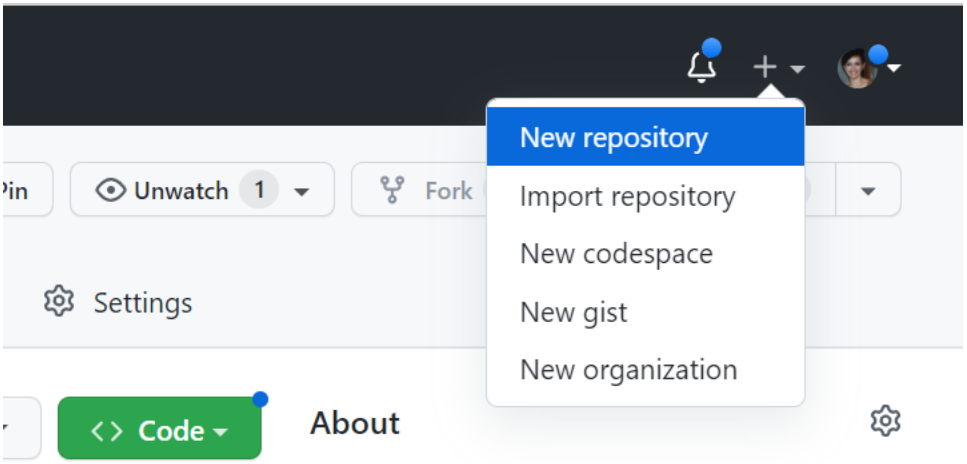
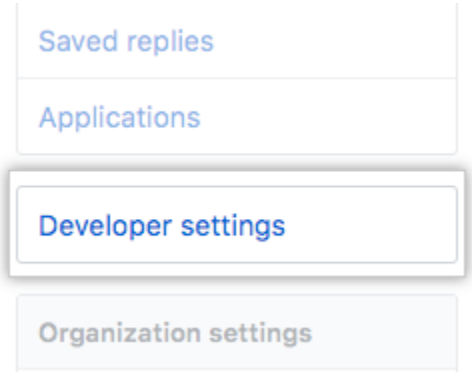
Lab Assignment # 5

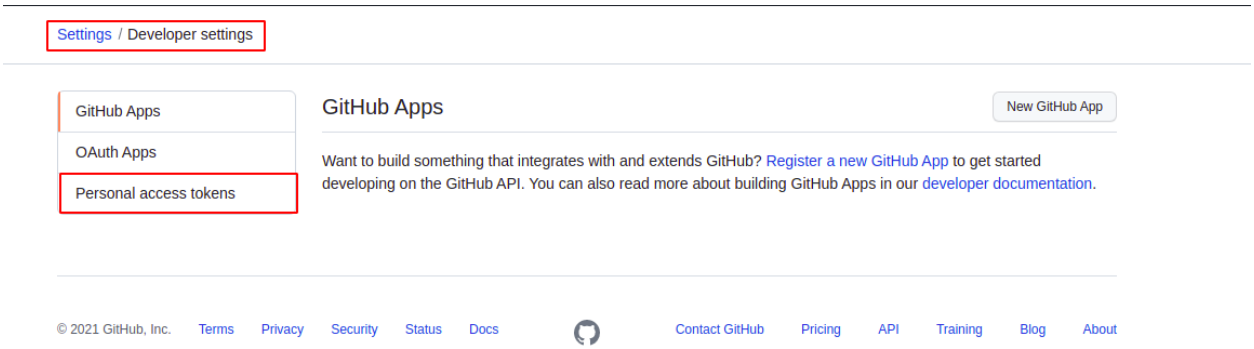
1. If you do not have a github account, sign up for a free https://github.com user account
2. Login to your account.
3. Create a repository with a README file – this file gives information about your repo, add details about how to build your repo. Add detailed description how to build your repository.



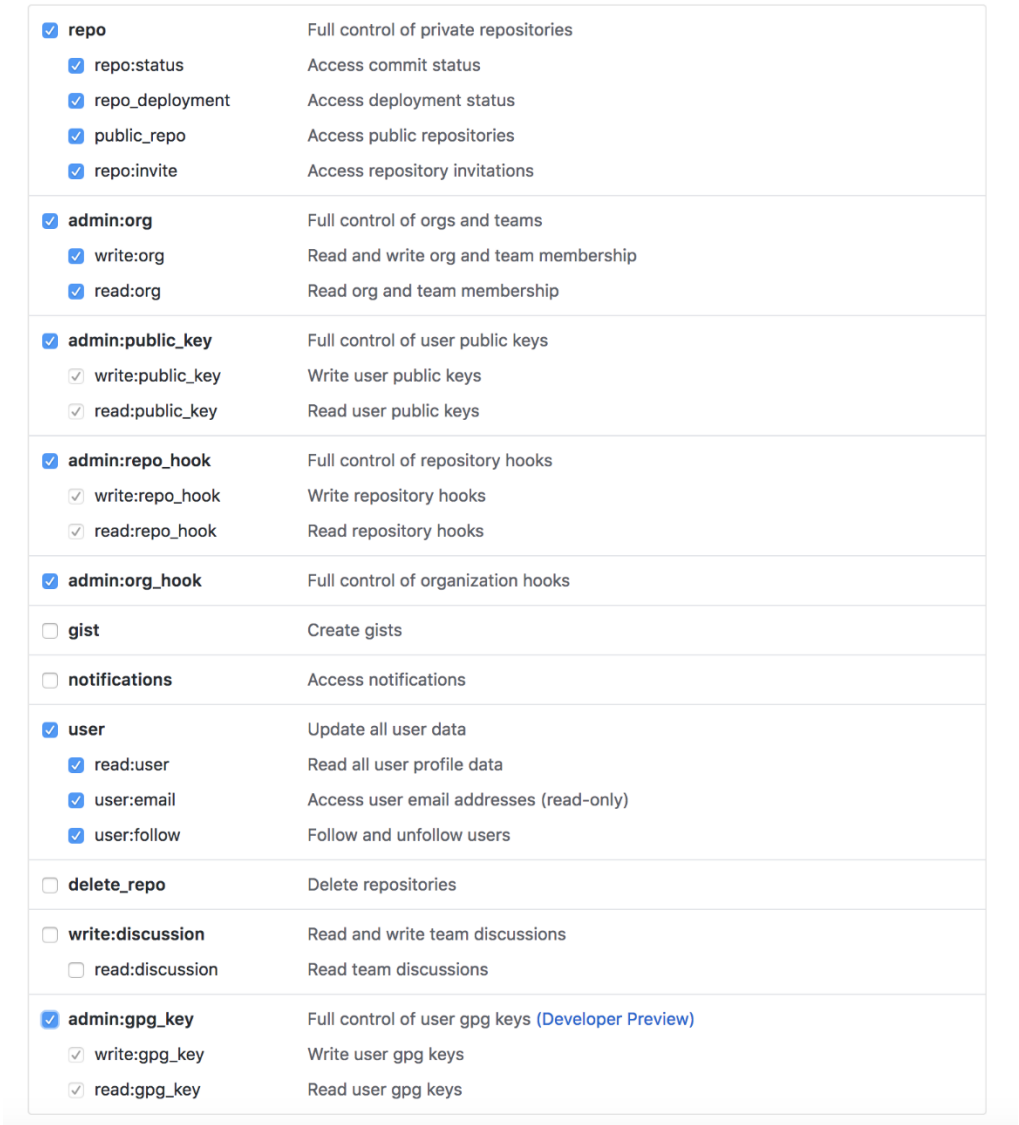
4. Now you have a repository at https://github.com/<YOUR_USER_NAME>/<YOUR_REPO_NAME>.git – this is your remote origin.
5. You can use git through command-prompt or powershell, however I would suggest using git-bash in Windows. I use git-bash at work and for my personal needs, it is a very good application. You can download at https://gitforwindows.org/.
6. Now go to your GitHub *Account settings.*
7. Click Developer Settings



8. Select Personal Access



9. Generate a token with the given permissions, e.g., This will generate a token, you will need to use this token at step 25.



10.Launch your terminal application (cmd-prompt, git-bash, terminal etc.)

11. Setup your user as follows:

```
git config --global user.name "YOUR NAME"
git config --global user.email your_email
```

12. Go to your local repo folder such as “C:\myRepos\”

- At git-bash you need to call cd /c/myRepos/

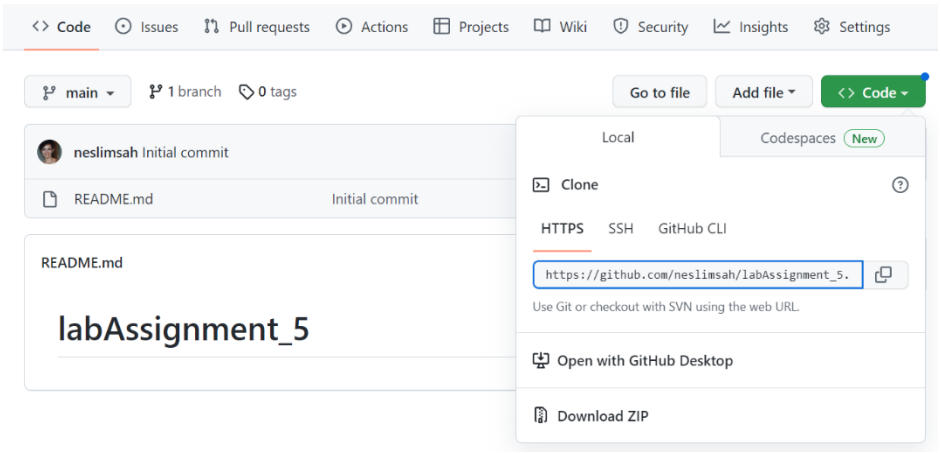
```
ntorosda@ORL-L10-TOROSDA MINGW64 /c/Users/ntorosda/Downloads
$ cd /c/myRepos/

ntorosda@ORL-L10-TOROSDA MINGW64 /c/myRepos
$ git clone https://github.com/neslimsah/labAssignment_5.git
Cloning into 'labAssignment_5'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 598 bytes | 12.00 KiB/s, done.

ntorosda@ORL-L10-TOROSDA MINGW64 /c/myRepos
$
```

13. Clone your remote repo to your PC:

call “`git clone` https://github.com/neslimsah/labAssignment_5.git”



14.Copy your completed lab_assignment_4 files to C:\myRepos\labassignment_5\ and rename them as lab_assignment_5.c and input.txt.

15. call “git status”

---- Take a screenshot of your output:

16. create a local working branch using “git checkout -b my_working_branch”

```
ntorosda@ORL-L10-TOROSDA MINGW64 /c/myRepos/labAssignment_5 (main)
$ git checkout -b my_working_branch
Switched to a new branch 'my_working_branch'

ntorosda@ORL-L10-TOROSDA MINGW64 /c/myRepos/labAssignment_5 (my_working_branch)
$
```

17. Add your updated files using “git add filename”. Always avoid using “git add .” because it causes untracked\unneeded files to be added, always add each file one by one.

18. Commit your staged updates using:

git commit

19. It will open an editor, write a meaningful commit message.

20. Save your changes, and close the editor.

21. Call “git log” to display your repo’s log history:

---- Take a screenshot of your output:

22. merge your local branch to the main branch:

```
ntorosda@ORL-L10-TOROSDA MINGW64 /c/myRepos/labAssignment_5 (my_working_branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

ntorosda@ORL-L10-TOROSDA MINGW64 /c/myRepos/labAssignment_5 (main)
$ git merge my_working_branch
Updating 702025b..9ff1015
Fast-forward
 input.txt          |  7 ++++++
 lab_assignment_5.c | 67 ++++++++++++++++++++++++++++++++++++++++++++++++++++++
 2 files changed, 74 insertions(+)
 create mode 100644 input.txt
 create mode 100644 lab_assignment_5.c

ntorosda@ORL-L10-TOROSDA MINGW64 /c/myRepos/labAssignment_5 (main)
$ git log
commit 9ff101504c7a5b6ec6b263c5a96c3c42590158ca (HEAD -> main, my_working_branch)
Author: Torosdagli <ntorosda@amd.com>
Date:   Fri Feb 10 22:18:46 2023 -0500

    Added lab_assignment_5.c and the input file.

commit 702025b96a359e11b647cb32d8d4bc67edeae1d5 (origin/main, origin/HEAD)
Author: Neslisah Torosdagli <neslisah.torosdagli@amd.com>
Date:   Fri Feb 10 21:32:04 2023 -0500

    Initial commit

ntorosda@ORL-L10-TOROSDA MINGW64 /c/myRepos/labAssignment_5 (main)
$
```

git checkout main

git merge my_working_branch

23. Display your git log history using “git log” in the main branch.

24. Push your updates to the remote origin using:

    git push

25. Provide a username and the generated token as a password
26. Your report in pdf format will be composed of:
    a. URL link of your github repository.
    b. Screenshots at steps 15 and 21.