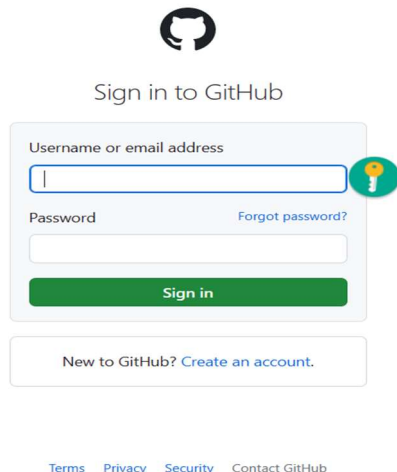


ASSIGNMENT 8

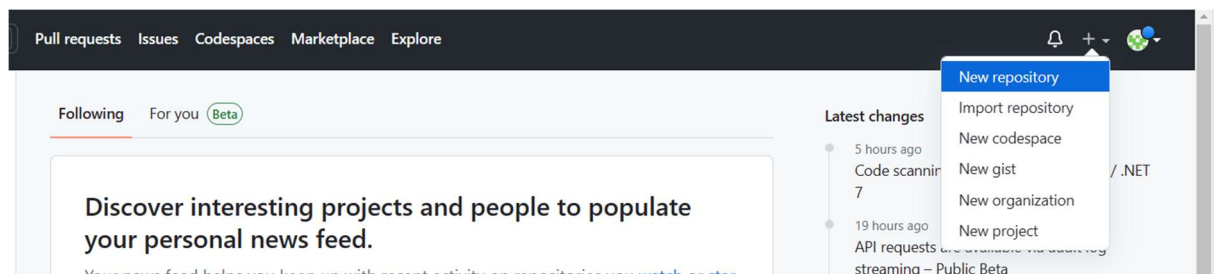
Problem Statement: *Deploy a project from local machine to Github and vice versa.*

CREATE NEW REPOSITORY

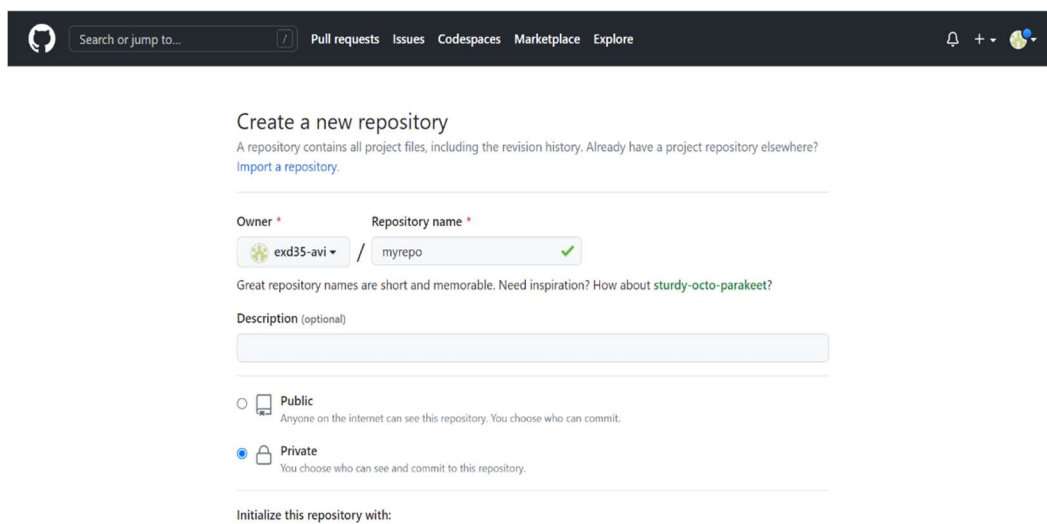
1. Sign in to your Github account.



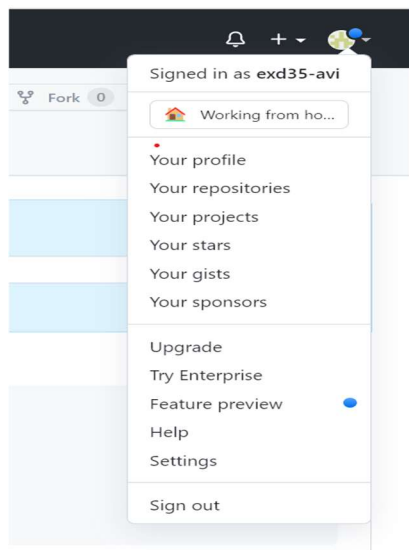
2. Click on the '+' sign in the top right then click on **New repository**.



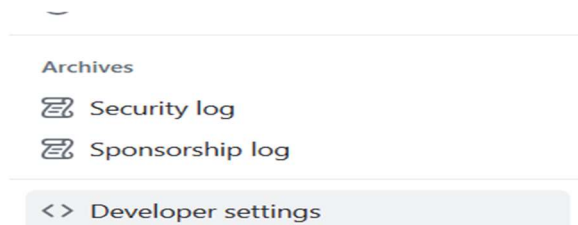
3. Give a **Repository name**. Next, select **Private** and click on **Create repository**.



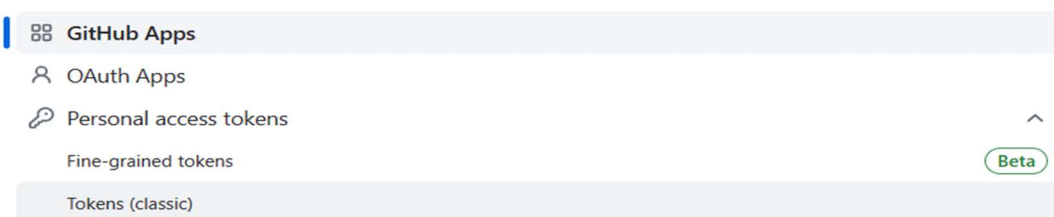
4. Click on your account at the top right, go to **Settings**.



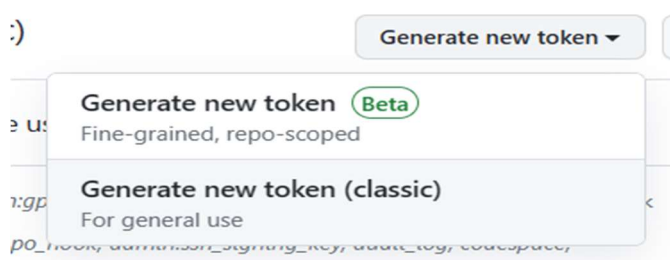
Then, go to Developer settings.



5. Select the drop-down menu of Personal access tokens and click **Tokens (classic)**.



6. Click **Generate new token**, then **Generate new token (classic)**.



7. Give the Token name, set **Expiration** days as 90 and select all the checkboxes. Click Generate token. Save your token.

Note

Token2

What's this token for?

Expiration *

90 days

The token will expire on Mon, Jul 3 2023

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input checked="" type="checkbox"/> read:org	Read org and team membership, read org projects
<input checked="" type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input checked="" type="checkbox"/> admin:public_key	Full control of user public keys
<input checked="" type="checkbox"/> write:public_key	Write user public keys
<input checked="" type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> admin:repo_hook	Full control of repository hooks

Continue...

Generate token

Cancel

- Go to the newly created repository and then to repository **Settings**. Click **Collaborators**. Then click **Add people** and invite people by searching.

The screenshot shows the GitHub repository settings page for 'exd35-avi / myrepo'. The repository is private. The 'Collaborators' tab is selected, showing a list of collaborators. The repository is set to 'PRIVATE' and 'Only the repository owner can push to this branch'. The 'Who to invite' section is visible on the right.

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/exd35-avi/myrepo.git>

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

General

Access

Collaborators

Code and automation

Actions

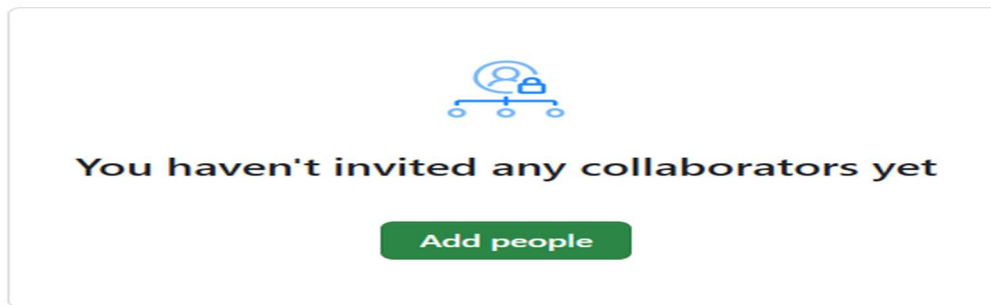
Who to invite

PRIVATE

Only the repository owner can push to this branch

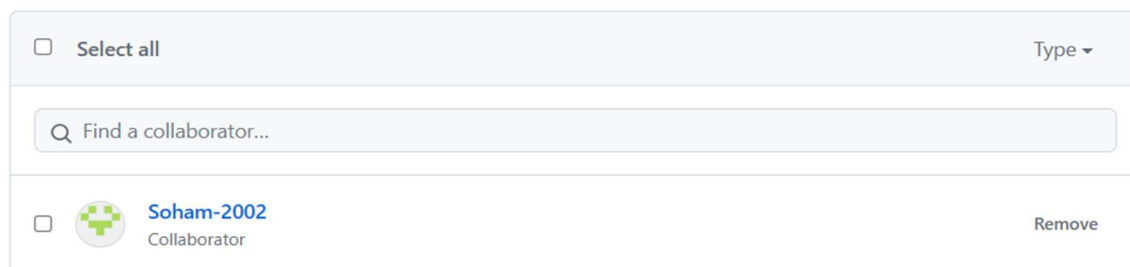
Manage

Manage access



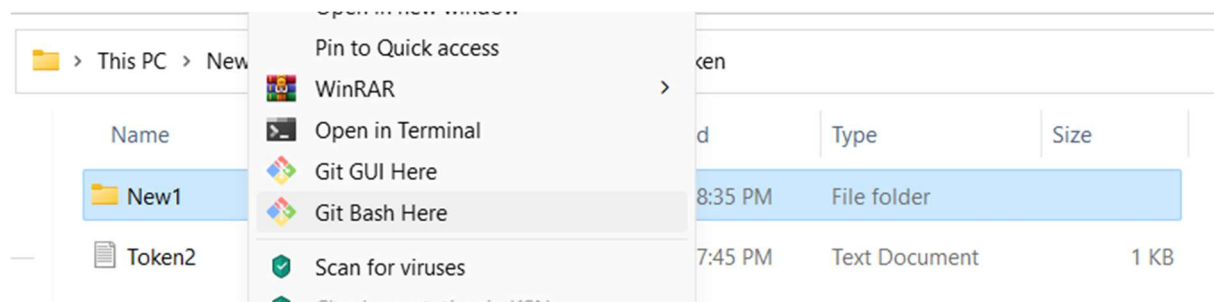
Manage access

Add people



DEPLOYING A PROJECT FROM LOCAL MACHINE TO GITHUB.

1. Open the required HTML folder with Git Bash.



2. Type and execute the following commands one by one.

- a. git init

```
aviru@AVIRUP MINGW64 /d/Avirup
$ git init
Initialized empty Git repository in D:/Avirup/.git/
```

- b. ls

```
aviru@AVIRUP MINGW64 /d/Avirup (master)
$ ls
about.html  avirupkey.pem  index.html  next.html
```

- c. git status

```
aviru@AVIRUP MINGW64 /d/Avirup (master)
$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will
  be committed)
    about.html
    avirupkey.pem
    index.html
    next.html
nothing added to commit but untracked files present
(use "git add" to track)
```

d. git add .

```
aviru@AVIRUP MINGW64 /d/Avirup (master)
$ git add .
warning: in the working copy of 'avirupkey.pem', LF
will be replaced by CRLF the next time Git touches
it
```

e. git status

```
aviru@AVIRUP MINGW64 /d/Avirup (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   about.html
        new file:   avirupkey.pem
        new file:   index.html
        new file:   next.html
```

f. git config --global user.email avirupmazumder82299@gmail.com

```
aviru@AVIRUP MINGW64 /d/Avirup (master)
$ git config --global user.email "avirupmazumder822
99@gmail.com"
```

g. git config --global user.name "exd35-avi"

```
aviru@AVIRUP MINGW64 /d/Avirup (master)
$ git config --global user.name "exd35-avi"
```

h. git commit -m "done"

```
aviru@AVIRUP MINGW64 /d/Avirup (master)
$ git commit -m "Done"
[master (root-commit) 877cb3d] Done
 4 files changed, 57 insertions(+)
 create mode 100644 about.html
 create mode 100644 avirupkey.pem
 create mode 100644 index.html
 create mode 100644 next.html
```

i. git remote add origin <https://github.com/exd35-avi/myrepo.git>

```
aviru@AVIRUP MINGW64 /d/Avirup (master)
$ git remote add origin https://github.com/exd35-av
i/myrepo.git
```

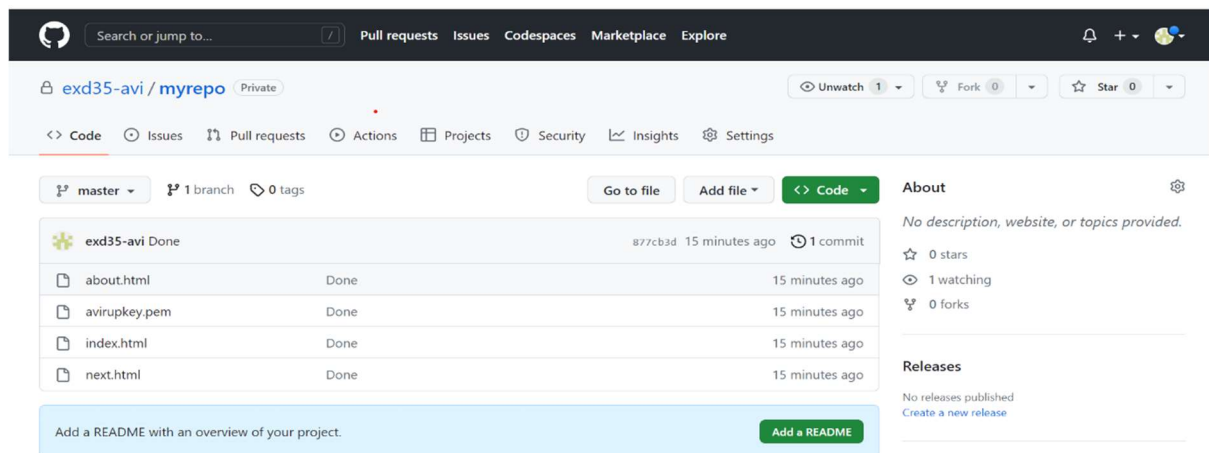
j. git push -u origin master

```

aviru@AVIRUP MINGW64 /d/Avirup (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.91 KiB | 1.91 MiB/s,
done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused
0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/exd35-avi/myrepo.git
* [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

```

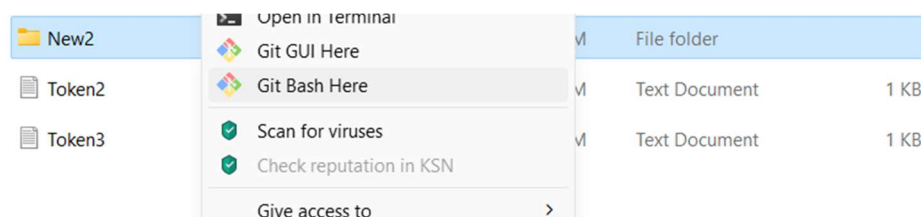
Go to your repository in Github and if it is already open, then refresh.



CLONING A GIVEN REPOSITORY FROM GITHUB TO LOCAL MACHINE AND UPLOADING THAT TO A NEWLY CREATED REPOSITORY

Cloning from Github to local machine

1. Create new directory and open Gitbash in this folder.



2. Type and execute the following commands

- a. git init

```

aviru@AVIRUP MINGW64 /d/New (master)
$ git init
Reinitialized existing Git repository in D:/New/.git/

```

- b. git clone <https://github.com/sudip7407/New-Repo1.git>

```

aviru@AVIRUP MINGW64 /d/New (master)
$ git clone https://github.com/sudip7407/New-Repo1.git
Cloning into 'third'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 15 (delta 6), reused 15 (delta 6), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (6/6), done.

```


Uploading the cloned project from local machine to my repository

1. Create a new repository in Github
2. Open Gitbash in the cloned folder (remove the already existing .git hidden folder in the folder containing the cloned project.)

Type and execute the following commands

- a. git init

```
aviru@AVIRUP MINGW64 /d/New/New-Repo1 (master)
$ git init
Reinitialized existing Git repository in D:/New/New-Repo1/.git/
```

- b. git add .

```
aviru@AVIRUP MINGW64 /d/New/New-Repo1 (master)
$ git add .
```

- c. git commit -m "Committed"

```
aviru@AVIRUP MINGW64 /d/New/New-Repo1 (master)
$ git commit -m "Committed"
On branch master
Your branch is up to date with 'origin/master'.
nothing to commit, working tree clean
```

- d. git remote add origin <https://github.com/exd35-avi/third.git>

```
aviru@AVIRUP MINGW64 /d/New/New-Repo1 (master)
$ git remote add origin https://github.com/exd35-avi/third.git
```

- e. git push -u origin master

```
aviru@AVIRUP MINGW64 /d/New/New-Repo1 (master)
$ git push -u origin master
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (15/15), 2.62 KiB | 2.62 MiB/s, done.
Total 15 (delta 6), reused 15 (delta 6), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/exd35-avi/third.git
* [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Open the repository and refresh it if it is already opened, now the clone is done.

The screenshot shows the GitHub interface for a repository named 'third' by user 'exd35-avi'. The repository is private and has 0 stars, 1 watching, and 0 forks. The main content area shows a commit by 'sudip7407' titled 'Update index.js' from 3 weeks ago, with 4 commits in total. The commit details show a list of files: .gitignore, New Text Document.txt, index.js, and package.json, all marked as 'done'. A button 'Add a README' is visible at the bottom. The right sidebar contains sections for 'About' (no description), 'Releases' (no releases published), and 'Packages'.