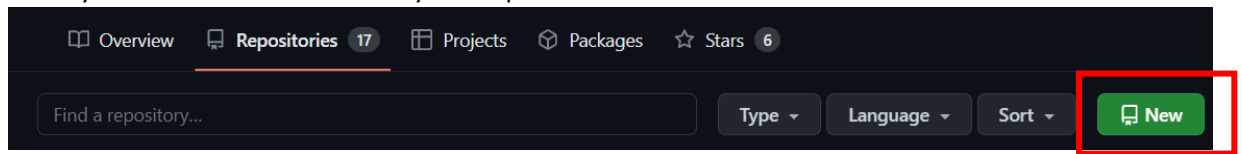


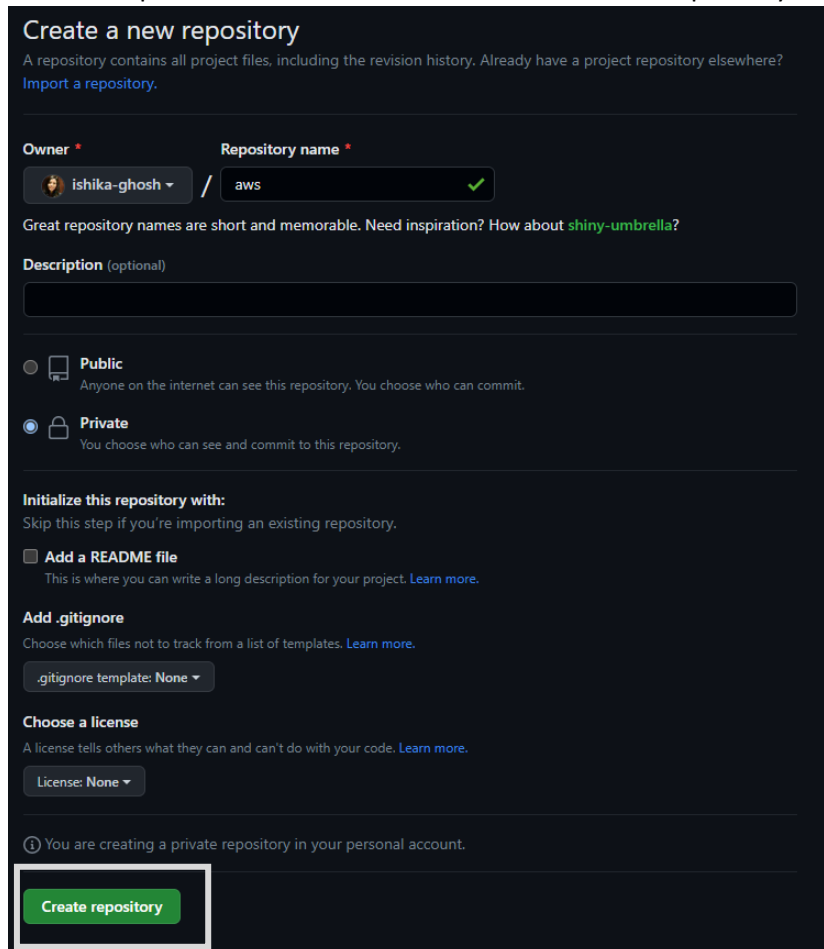
Assignment 8: Deploy a project from the local machine to GitHub and vice versa.

Create new repository

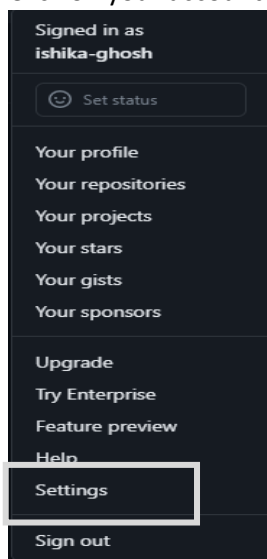
1. Go to your Github Account. Go to your Repositories. Click New.



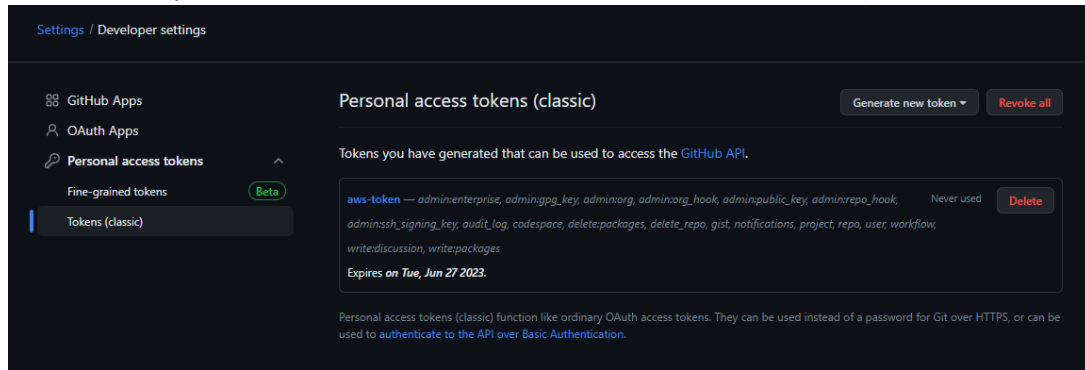
2. Give the repo name. Select Private and click on Create Repository.

A screenshot of the 'Create a new repository' form on GitHub. The form has a title 'Create a new repository' and a subtitle 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. The 'Owner' field is set to 'ishika-ghosh' and the 'Repository name' field is set to 'aws' with a green checkmark. Below these fields is a note: 'Great repository names are short and memorable. Need inspiration? How about [shiny-umbrella?](#)'. The 'Description (optional)' field is empty. The 'Visibility' section shows 'Public' and 'Private' options, with 'Private' selected. Below this is the 'Initialize this repository with:' section, which includes a checkbox for 'Add a README file' and a dropdown for '.gitignore template' set to 'None'. The 'Choose a license' section has a dropdown set to 'None'. At the bottom, a green 'Create repository' button is highlighted with a white rectangular box. A note at the bottom states: 'You are creating a private repository in your personal account.'

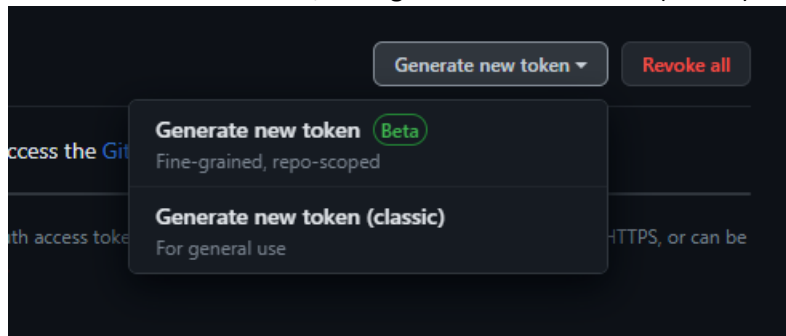
3. Click on your account at the top right, go to Settings then go to Developer settings.



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



5. Click Generate new token, then generate a new token (classic).



6. Give the token name, set expiration days as 90, and select all the checkboxes. Click Generate token. Save your token.

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

token1

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
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks
<input checked="" type="checkbox"/> read:repo_hook	Read repository hooks
<input checked="" type="checkbox"/> admin:org_hook	Full control of organization hooks
<input checked="" type="checkbox"/> gist	Create gists
<input checked="" type="checkbox"/> notifications	Access notifications
<input checked="" type="checkbox"/> user	Update ALL user data
<input checked="" type="checkbox"/> read:user	Read ALL user profile data
<input checked="" type="checkbox"/> user:email	Access user email addresses (read-only)
<input checked="" type="checkbox"/> user:follow	Follow and unfollow users
<input checked="" type="checkbox"/> delete_repo	Delete repositories
<input checked="" type="checkbox"/> write:discussion	Read and write team discussions
<input checked="" type="checkbox"/> read:discussion	Read team discussions
<input checked="" type="checkbox"/> admin:enterprise	Full control of enterprises
<input checked="" type="checkbox"/> manage_runners:enterprise	Manage enterprise runners and runner groups
<input checked="" type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input checked="" type="checkbox"/> read:enterprise	Read enterprise profile data
<input checked="" type="checkbox"/> audit_log	Full control of audit log
<input checked="" type="checkbox"/> read:audit_log	Read access of audit log
<input checked="" type="checkbox"/> codespace	Full control of codespaces
<input checked="" type="checkbox"/> codespace:secrets	Ability to create, read, update, and delete codespace secrets
<input checked="" type="checkbox"/> project	Full control of projects
<input checked="" type="checkbox"/> read:project	Read access of projects

<input checked="" type="checkbox"/> admin:gpg_key	Full control of public user GPG keys
<input checked="" type="checkbox"/> write:gpg_key	Write public user GPG keys
<input checked="" type="checkbox"/> read:gpg_key	Read public user GPG keys
<input checked="" type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input checked="" type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input checked="" type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

[Generate token](#)

[Cancel](#)

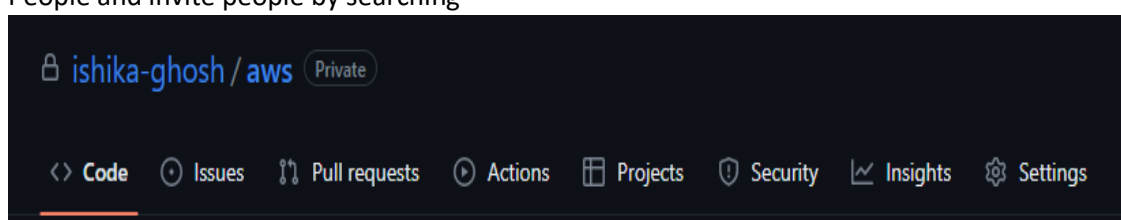
Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ [Copy](#) [Delete](#)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

- Go to the newly created repository and then to repository settings. Click Collaborators. Select Add People and invite people by searching



General

Access

Collaborators

Code and automation

Actions

Webhooks

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

Manage

DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

Add people

Add a collaborator to aws

Purnadip Manna

purnadip-manna

Add purnadip-manna to this repository

Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

Manage

DIRECT ACCESS

1 has access to this repository. 0 collaborators, 1 invitation.

Manage access

Add people

Select all

Type

Find a collaborator...

Purnadip Manna

Awaiting purnadip-manna's response

Pending Invite

Remove

Deploying a project from a local machine to GitHub and vice versa

1. Open the required html folder with Git Bash.
2. Type and execute the following commands one by one.
 1. `git init`
 2. `ls -l`
 3. `git status`
 4. `git add .`
 5. `git status`
 6. `git commit -m "first commit"`
 7. `git remote add origin repo link`
 8. `git push-u origin master`

```
ISHIKA@LAPTOP-2OPEKLT MINGW64 ~/Desktop/ishika2
$ git init
Initialized empty Git repository in C:/Users/ISHIKA/Desktop/ishika2/.git/

ISHIKA@LAPTOP-2OPEKLT MINGW64 ~/Desktop/ishika2 (master)
$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
ISHIKA@LAPTOP-2OPEKLT MINGW64 ~/Desktop/ishika2 (master)
$ git add .

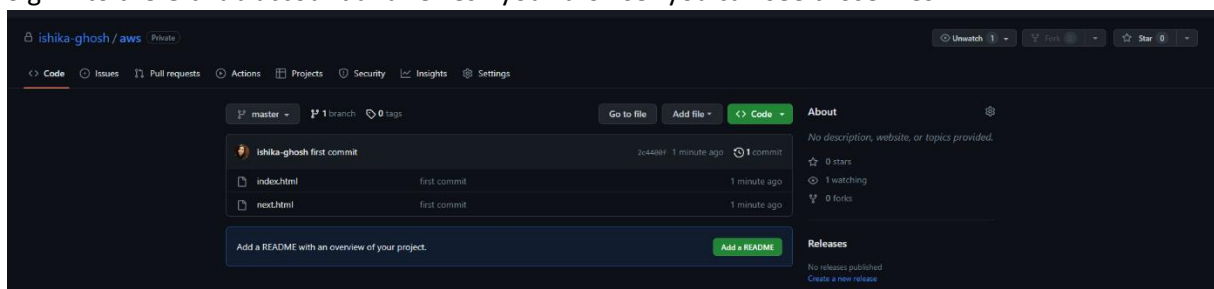
ISHIKA@LAPTOP-2OPEKLT MINGW64 ~/Desktop/ishika2 (master)
$ git commit -m "first commit"
[master (root-commit) 2c4400f] first commit
 2 files changed, 29 insertions(+)
 create mode 100644 index.html
 create mode 100644 next.html

ISHIKA@LAPTOP-2OPEKLT MINGW64 ~/Desktop/ishika2 (master)
$ git remote add origin https://github.com/ishika-ghosh/aws.git

ISHIKA@LAPTOP-2OPEKLT MINGW64 ~/Desktop/ishika2 (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 560 bytes | 560.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/ishika-ghosh/aws.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

ISHIKA@LAPTOP-2OPEKLT MINGW64 ~/Desktop/ishika2 (master)
```

3. Sign into the GitHub account and refresh your browser you can see those files.



4. Create a new directory and open Gitbash in this folder.
5. Type and execute the following commands
 1. `git clone https://github.com/sudip7407/New-Repo1.git`
6. Create a new repository in GitHub
7. Open Gitbash in the cloned folder (remove the already existing `.git` hidden folder in the folder containing the cloned project.)
8. Type and execute the following commands
 1. `git init`
 2. `git add .`

3. `git commit -m "Committed"`
 4. `git remote add origin repo link`
 5. `git push -u origin master`
9. Refresh your repository.

```
ISHIKA@LAPTOP-2OPEKLT V MINGW64 ~/Desktop/new
$ git clone https://github.com/sudip7407/New-Repo1.git
Cloning into 'New-Repo1'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 6), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (6/6), done.

ISHIKA@LAPTOP-2OPEKLT V MINGW64 ~/Desktop/new
$ git remote --v
fatal: not a git repository (or any of the parent directories): .git

ISHIKA@LAPTOP-2OPEKLT V MINGW64 ~/Desktop/new
$ cd New-Repo1/

ISHIKA@LAPTOP-2OPEKLT V MINGW64 ~/Desktop/new/New-Repo1 (master)
$ git remote --v
origin https://github.com/sudip7407/New-Repo1.git (fetch)
origin https://github.com/sudip7407/New-Repo1.git (push)

ISHIKA@LAPTOP-2OPEKLT V MINGW64 ~/Desktop/new/New-Repo1 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

ISHIKA@LAPTOP-2OPEKLT V MINGW64 ~/Desktop/new/New-Repo1 (master)
$ git remote add origin https://github.com/ishika-ghosh/awsproject.git
error: remote origin already exists.

ISHIKA@LAPTOP-2OPEKLT V MINGW64 ~/Desktop/new/New-Repo1 (master)
$ git remote remove origin

ISHIKA@LAPTOP-2OPEKLT V MINGW64 ~/Desktop/new/New-Repo1 (master)
$ git remote add origin https://github.com/ishika-ghosh/awsproject.git

ISHIKA@LAPTOP-2OPEKLT V MINGW64 ~/Desktop/new/New-Repo1 (master)
$ git push -u origin master
Everything up-to-date
Branch 'master' set up to track remote branch 'master' from 'origin'.

ISHIKA@LAPTOP-2OPEKLT V MINGW64 ~/Desktop/new/New-Repo1 (master)
```

