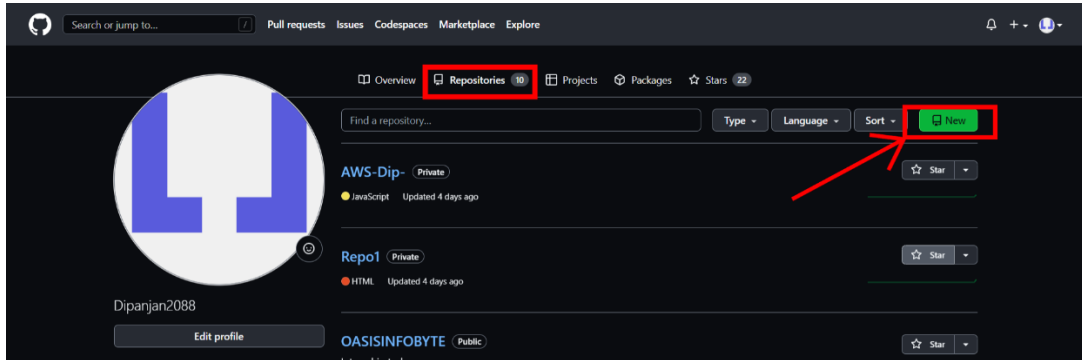


Assignment 8

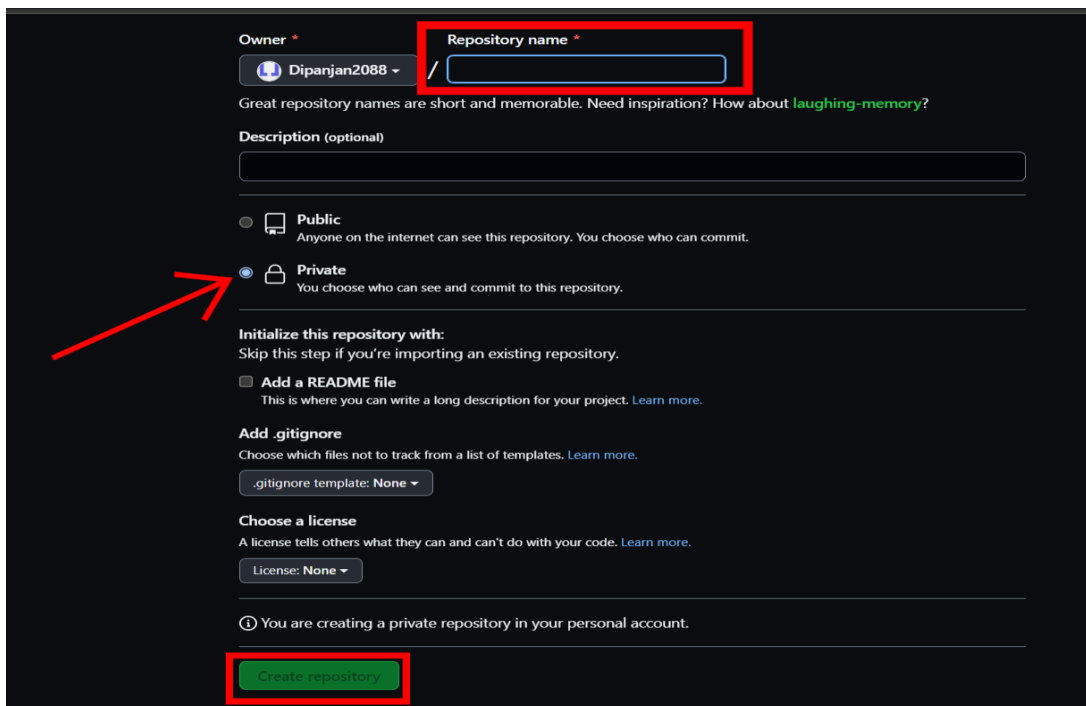
Deploy a project through Git.

Steps for deploying project through git:

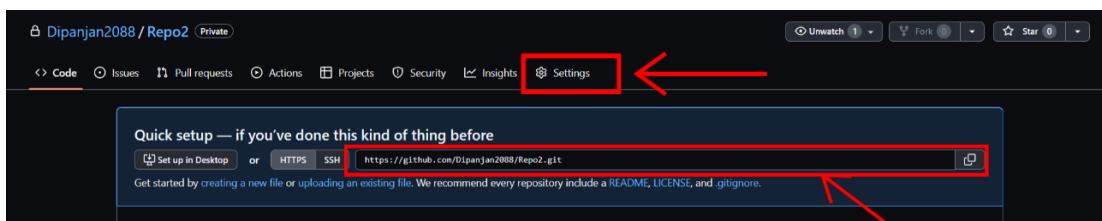
1. **Sign in.** Sign in to your GitHub account if have or just create one.
2. Go to **repository** and click on **new**.



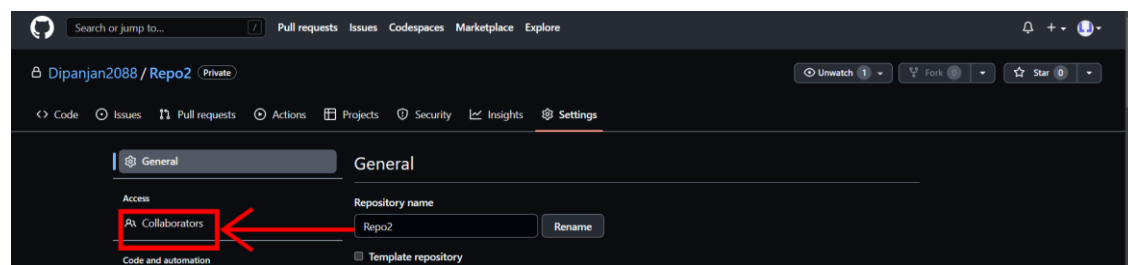
3. Enter **repository name**, make it **private** after that click on **create repository**.



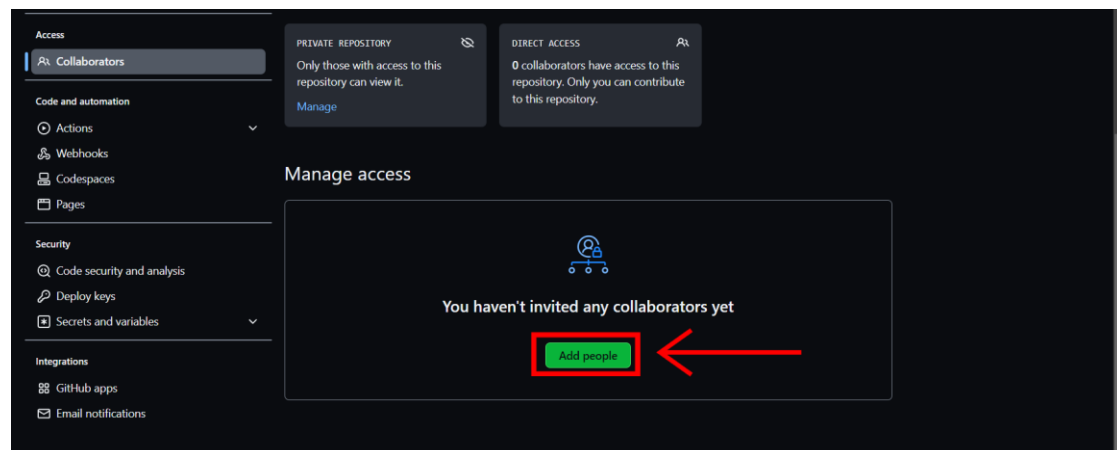
4. After that copy the **repository link** and **save it**. After that click on **settings**.



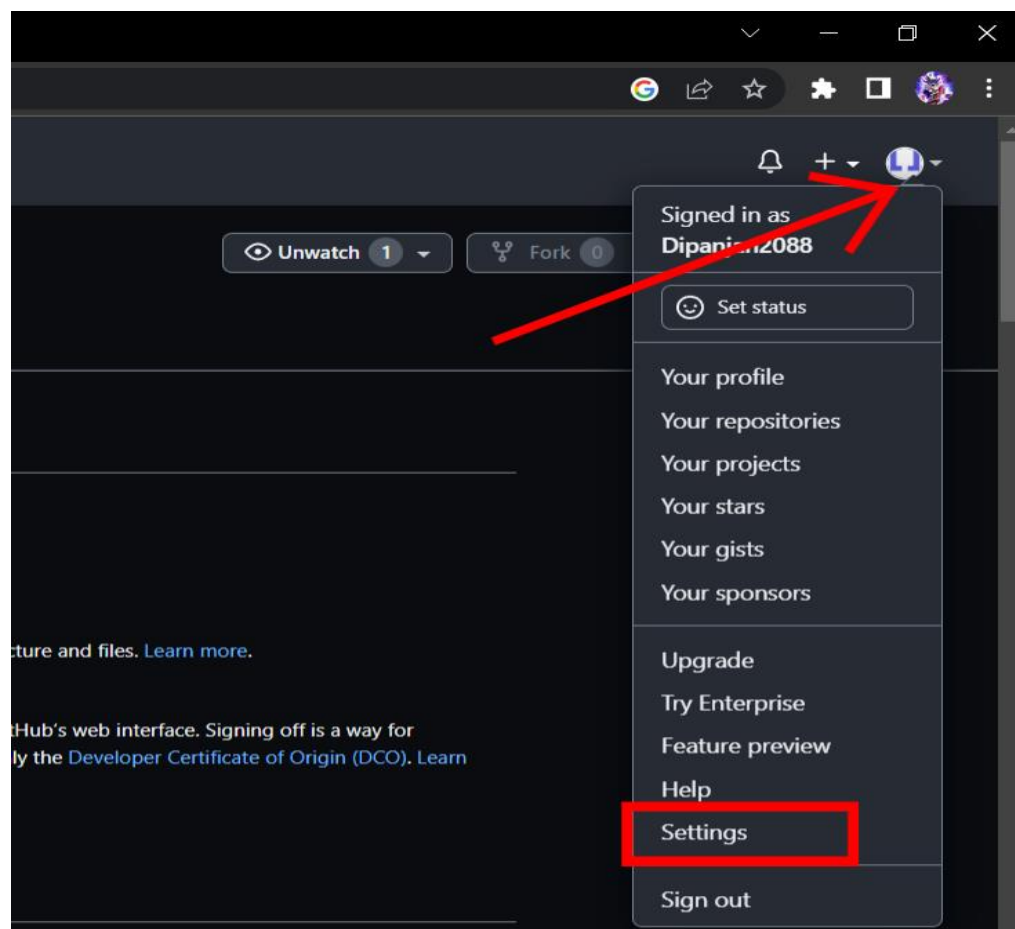
5. Click on **collaborators**.



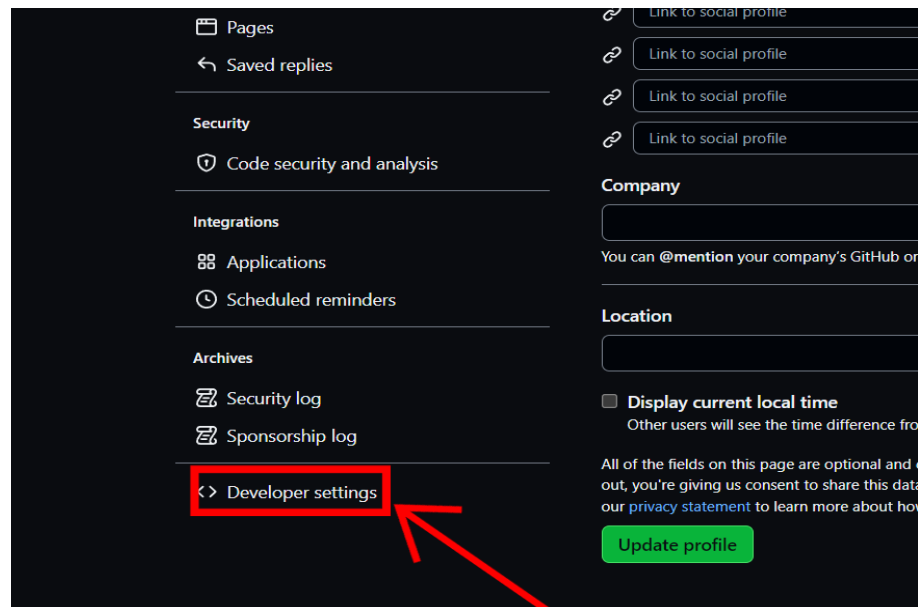
6. Click on **Add people** to invite or add your friends as a collaborator of your project.



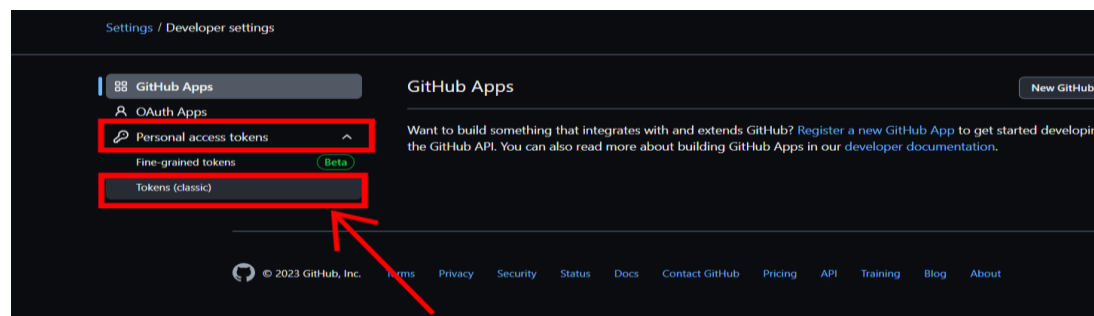
7. Click on your **profile icon**, in the drop down click on **settings**.



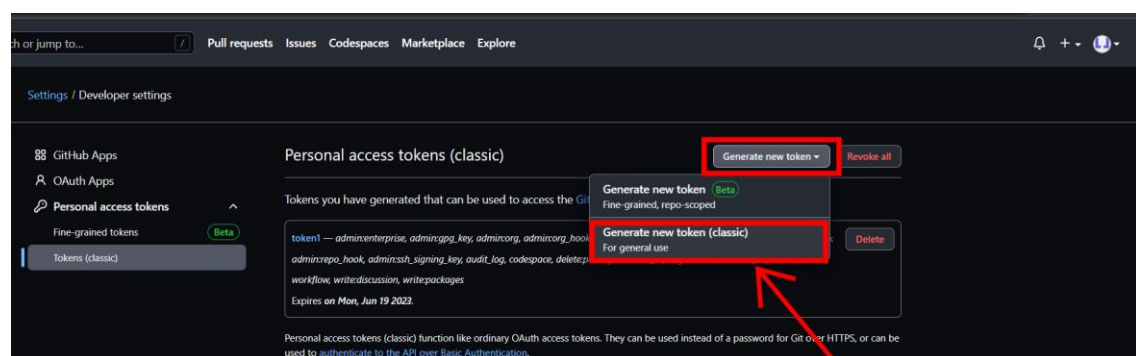
8. On the left side of the screen **scroll down** and click on **Developer settings**.



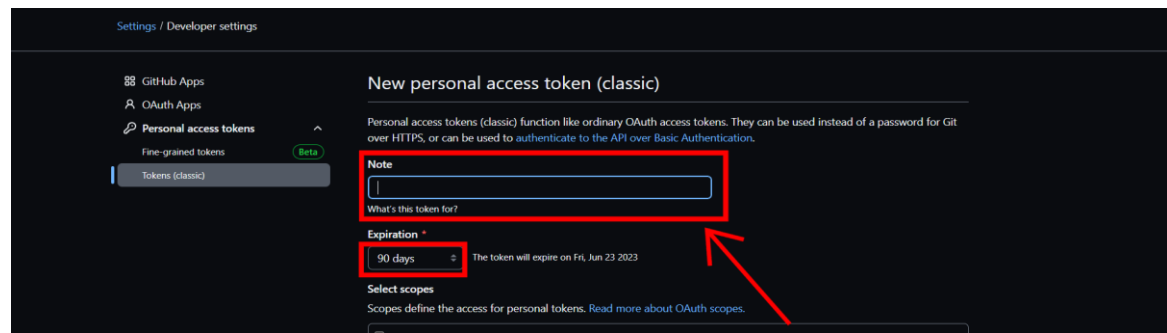
9. After that click on **Personal access token** and in Personal access token click on **Token (classic)**.



10. Click on **Generate new token** and Click on **Generate new token (classic)**.



11. Enter the **note of the token**. After that set the **Expiration**.



Settings / Developer settings

GitHub Apps
OAuth Apps
Personal access tokens
Fine-grained tokens (Beta)
Tokens (classic)

New personal access token (classic)

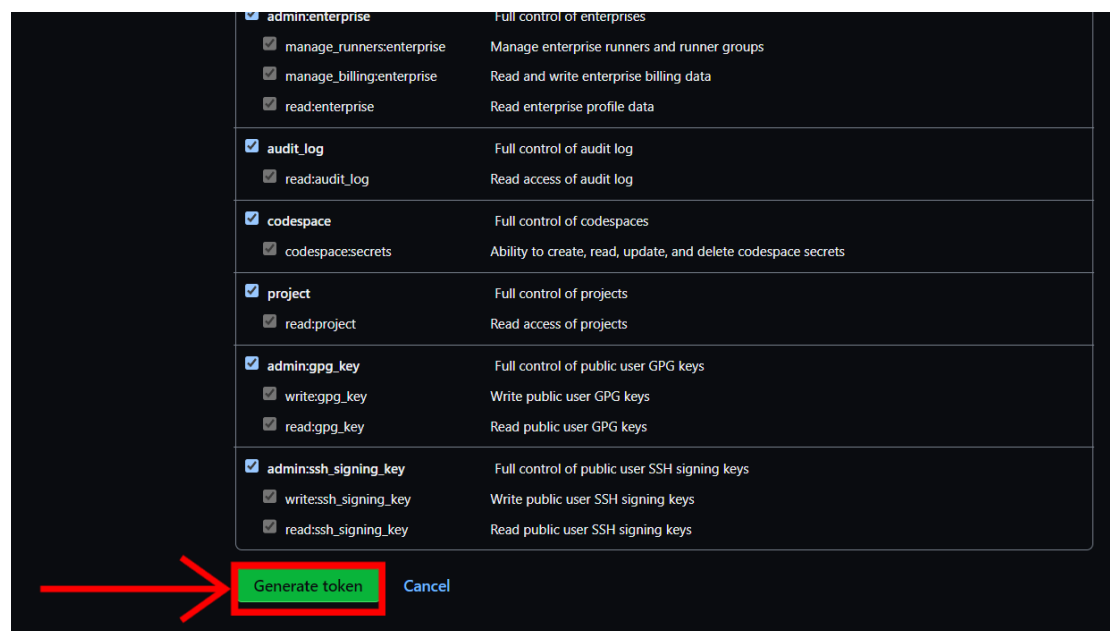
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note
What's this token for?

Expiration *
90 days The token will expire on Fri, Jun 23 2023

Select scopes
Scopes define the access for personal tokens. Read more about OAuth scopes.

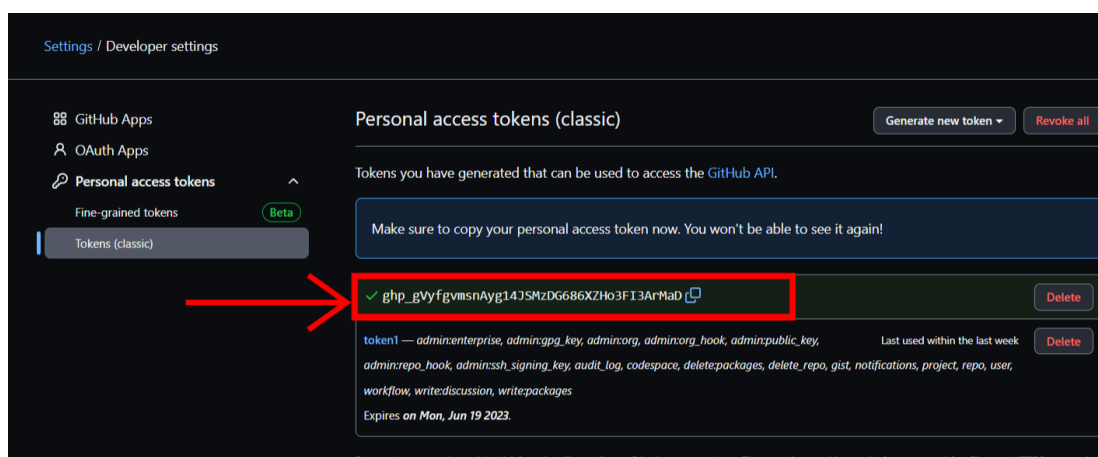
12. Click **all the check box** and click on **Generate token**.



<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:pgp_key	Full control of public user GPG keys
<input checked="" type="checkbox"/> write:pgp_key	Write public user GPG keys
<input checked="" type="checkbox"/> read:pgp_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

13. Copy the **token id** and save it.



Settings / Developer settings

Personal access tokens (classic)

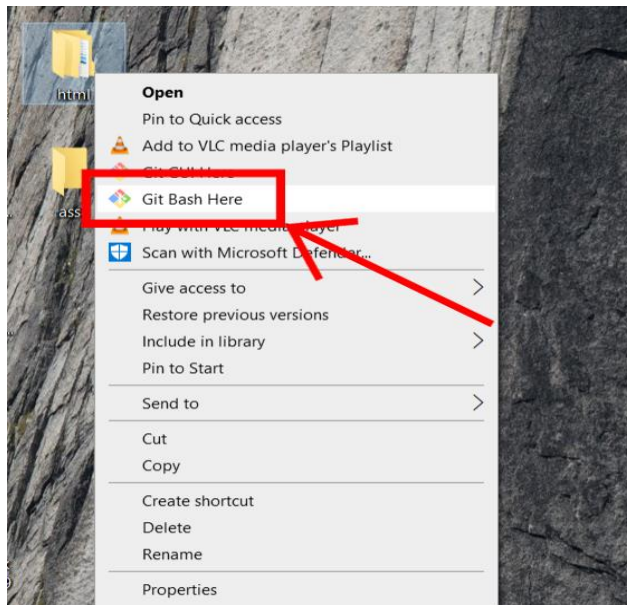
Generate new token Revoke all

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!

✓ 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	Delete
token1 — admin:enterprise, admin:pgp_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:packages Expires on Mon, Jun 19 2023. Last used within the last week	

14. **Right click** on the folder you want to upload in github. After that click on **Git Bash Here**.



15. After that enter the commands to upload the files on github.

- i) **git init** command create a new git repository. It can be used to convert an existing, un-versioned project to a Git repository or initialize a new, empty repository

```
ASUS@DESKTOP-IGCDJSK MINGW64 ~/Desktop/html
$ git init
Initialized empty Git repository in C:/Users/ASUS/Desktop/html/.git/
ASUS@DESKTOP-IGCDJSK MINGW64 ~/Desktop/html (master)
$
```

- ii) **git config --global user.email "Your github email ID"** is use for connecting with your Github account.

```
ASUS@DESKTOP-IGCDJSK MINGW64 ~/Desktop/html (master)
$ git config --global user.email "dipanjanmahata13@gmail.com"
ASUS@DESKTOP-IGCDJSK MINGW64 ~/Desktop/html (master)
$
```

- iii) **git add .** Command adds a change in the working directory to the staging area.

```
ASUS@DESKTOP-IGCDJSK MINGW64 ~/Desktop/html (master)
$ git add .
```

iv) **git commit -m "done"** : The -m option of commit command lets you to write the commit message on the command line.

```
ASUS@DESKTOP-IGCDJSK MINGW64 ~/Desktop/html (master)
$ git commit -m "done"
[master (root-commit) 23cce74] done
3 files changed, 21 insertions(+)
create mode 100644 index.html
create mode 100644 next.html
create mode 100644 third.html
```

v) **git remote add origin *remote link ***: To add a new remote, use the git remote add command on the terminal, in the directory your repository is stored at.

***instead of writing Remote like paste the link you copied in step 4.**

```
ASUS@DESKTOP-IGCDJSK MINGW64 ~/Desktop/html (master)
$ git remote add origin https://github.com/Dipanjan2088/Repo2.git
```

vi) **git push -u origin master** command can be used to push any commits made locally on the `master` branch to a remote repository on `origin`.

```
ASUS@DESKTOP-IGCDJSK MINGW64 ~/Desktop/html (master)
$ git push -u origin master
```

16. After this command you will get a popup where you have to paste the token which you have copied in step 13.

17. Now go to the repository you created and you can see that all the files are uploaded on your github repository.

The screenshot shows the GitHub interface for a repository named 'Dipanjan2088/Repo2'. The repository is private and has 0 stars, 1 watching, and 0 forks. The main content area shows a commit by 'Dipanjan2088' with the message 'done' at commit hash '23cce74', made 5 minutes ago. The commit details show three files: 'index.html', 'next.html', and 'third.html', all marked as 'done' and created 5 minutes ago. There is a button to 'Add a README' with an overview of the project. The right sidebar contains the 'About' section with a note 'No description, website, or topics provided.' and the 'Releases' section with a note 'No releases published' and a button to 'Create a new release'.