

Dear Students

Today lets learn how to work with git.

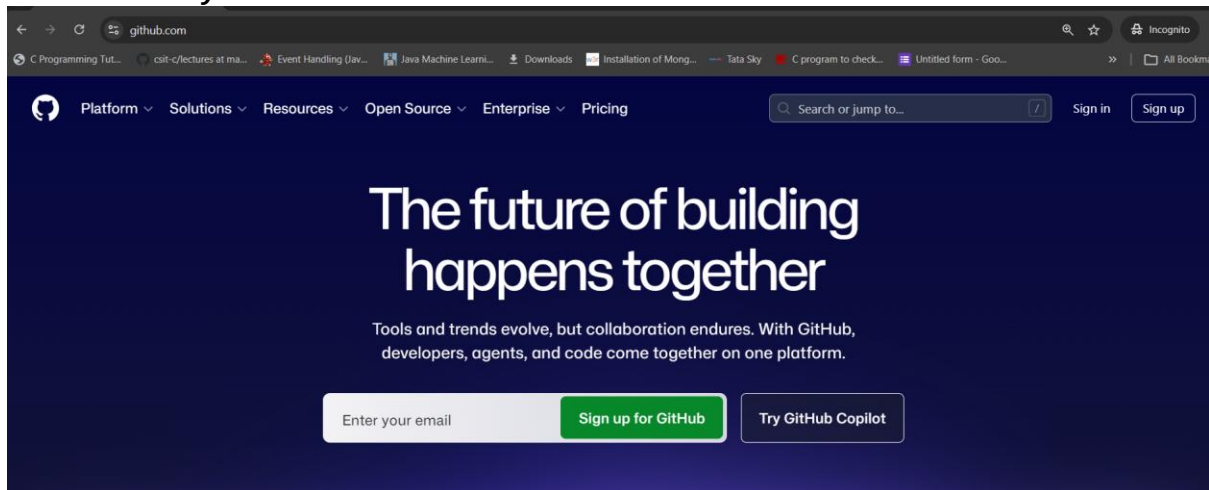
Step 1: What is Git

Git is a version control subsystem to manage and collaborate the code among multiple developers in a project while developing any application.

Step 2: How to use?

So to use git students you need to create a github account.

Let me show you now.



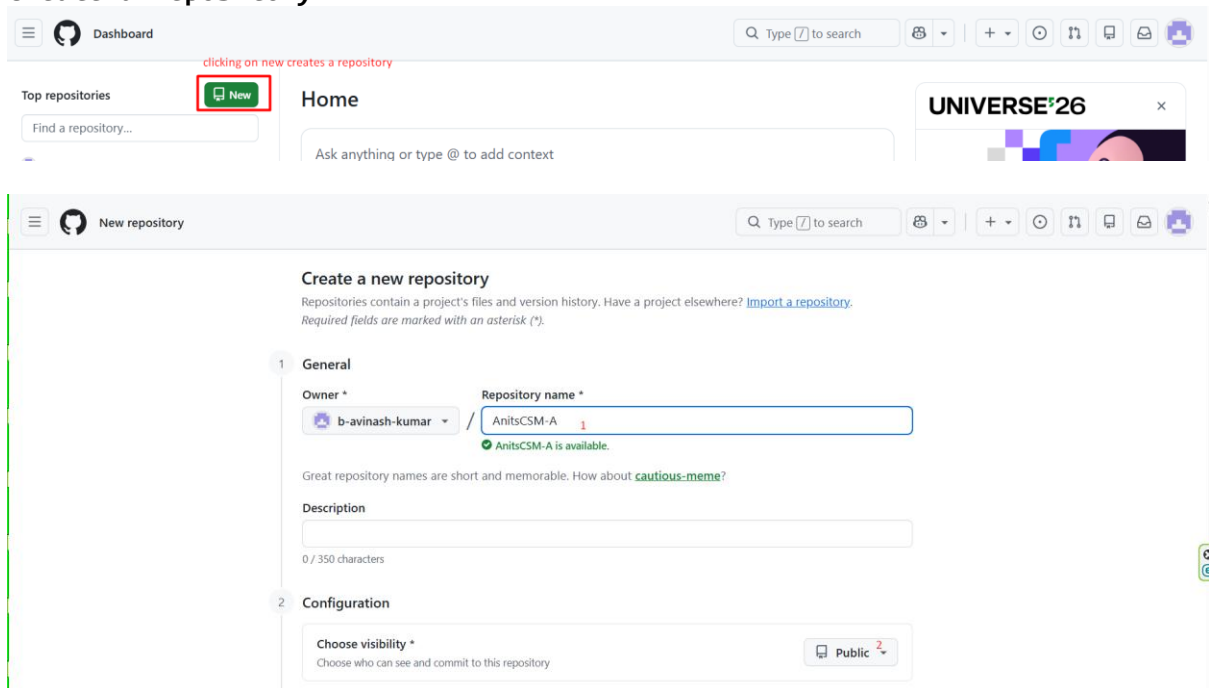
Signup

After signing up you can verify through your email sent by github

After email verification

Log in

Create a repository



Description

0 / 350 characters

2 Configuration

Choose visibility *
Choose who can see and commit to this repository Public

Add README
READMEs can be used as longer descriptions. [About READMEs](#) Off

Add .gitignore
.gitignore tells git which files not to track. [About ignoring files](#) No .gitignore

Add license
Licenses explain how others can use your code. [About licenses](#) No license

3 Create repository

© 2025 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Community](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

By following above steps you will be able to create Github account and create repository upon login.

Now lets go to repository that we created named as AnitsCSM-A

b-avinash-kumar / AnitsCSM-A

Code Issues Pull requests Actions Projects Wiki Security and quality Insights Settings

AnitsCSM-A Public Pin Watch 0 Fork 0 Star 0

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

Add collaborators to this repository
Search for people using their GitHub username or email address.
[Invite collaborators](#)

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

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/b-avinash-kumar/AnitsCSM-A.git>

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

...or create a new repository on the command line

```
echo "# AnitsCSM-A" >> README.md
```

Repository got created

Now you have a doubt how to validate weather repository created or not?
For this you need to go back to github in web browser.

b-avinash-kumar / AnitsCSM-A

Code Issues Pull requests Actions Projects Wiki Security and quality Insights Settings

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

1. Repository link

Set up in Desktop or **HTTPS** SSH <https://github.com/b-avinash-kumar/AnitsCSM-A.git>

3. As we already created repository in github web way we need to use git clone repoU for synching remote repo with local repo.

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

...or create a new repository on the command line

2. On your machine set of commands to be used to create repository manu:

Note: Don't use this nk

```
echo "# AnitsCSM-A" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/b-avinash-kumar/AnitsCSM-A.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/b-avinash-kumar/AnitsCSM-A.git
git branch -M main
git push -u origin main
```

Now you may have a doubt what is remote repository?

A remote repository is the location where we created our repository named "AnitsCSM-A" at github servers using github.com.

Clear

C:\>git clone repositoryURL

e.g. git clone <https://github.com/b-avinash-kumar/AnitsCSM-A.git>

hit enter you will be able to see .git hidden folder.

After creating git recognizable directory on your local machine, now you need to create your codes i.e write codes and update the files/folders, code which are created to github remote repository (remote location).

Now at your local machine you need to install a gitbash software then only you can work with your repository.

Download the gitbash

 gitbash

AI Mode **All** Images Videos Shopping Forums News More ▾ Tools ▾

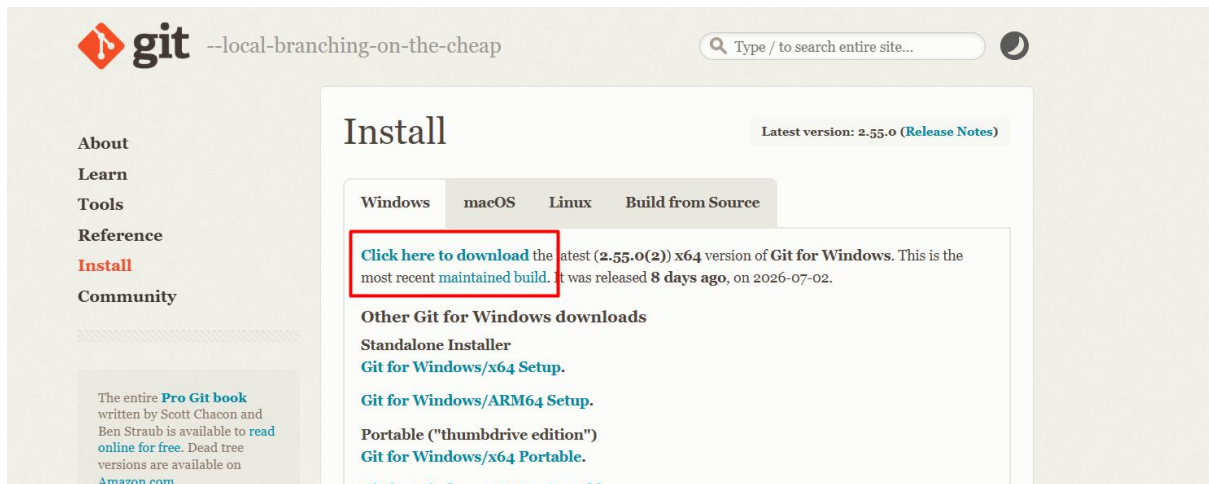
These are results for **git bash**

Search instead for **gitbash**

 **Git**
<https://git-scm.com> › install › windows

Git - Install for Windows

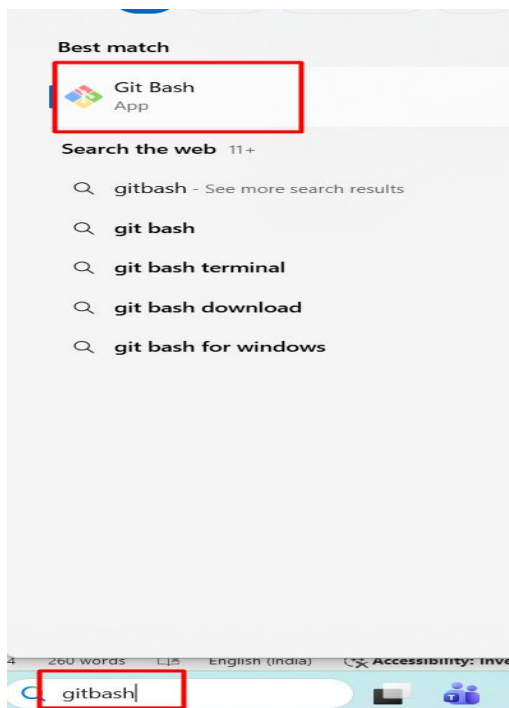
Click here to download the latest (2.55.0(2)) x64 version of **Git** for Windows. This is the most recent maintained build. It was released 8 days ago, ... [Read more](#)



After downloading the gitbash install the software.

Once the software got installed you need to open git bash or command prompt or vs code terminal window to work with git remote repository.

Clear.



Now lets work in vs code

```
<!--
```

add the file index.html to remote repository

```
git add .
```

index.html will be moving from working directory to staged area

```
git commit -m 'added index page'
```

it moves index.html file from staged area to local repository on your machine.

and from here we can send to remote repository

```
git push
```

as a beginner for the first time you need to establish trust
ok
then how?
Go to github
-->

Account -> Settings -> Developer Settings -> Tokens(Classic) -> Generate
classic token -> verify with code sent to email -> Name the token ->
select the repo checkbox

☒ repo	Full control of private repositories
☒ repo:status	Access commit status
☒ repo_deployment	Access deployment status
☒ public_repo	Access public repositories
☒ repo:invite	Access repository invitations
☒ security_events	Read and write security events
☐ read:network_configurations	Read org hosted compute network configurations
☐ project	Full control of projects
☐ read:project	Read access of projects
☐ admin:gpg_key	Full control of public user GPG keys
☐ write:gpg_key	Write public user GPG keys
☐ read:gpg_key	Read public user GPG keys
☐ admin:ssh_signing_key	Full control of public user SSH signing keys
☐ write:ssh_signing_key	Write public user SSH signing keys
☐ read:ssh_signing_key	Read public user SSH signing keys

[Generate token](#) [Cancel](#)

Then hit generate

Now token created is displayed you need to copy and save it and token will be expired after 30days.

Note: If you forget to save the token and try to revisit the created token the token will not be available so save after creation of token.

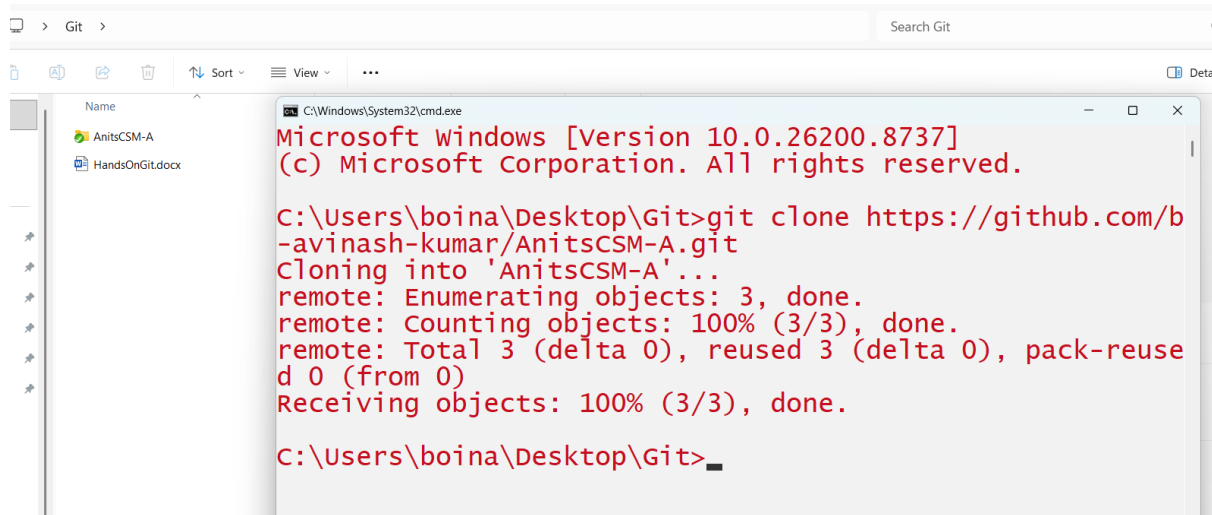
Clear

Add = add the files from your working directory to staged area.
Commit = add the files from staged area to local repository
Push = add the files from local repository to remote repository

Note:

Until commit operation trust between local repository and remote repository is not required.

For the first time as a new user when you want to push the files from local repository to remote repository then for the first time we need to establish trust using git token that we generated.



After clone change directory to git repository folder

Suppose you created an index.html and other files these files first we need to add in staged area using “git add .”

`git add .`

optional step : check the status using “git status” if you want.

Now commit the files in staged area to local repository

`git commit -m 'message'`

```
C:\Users\boina\Desktop\Git\AnitsCSM-A>git commit -m "first commit"
On branch main
Your branch is up to date with 'origin/main'.

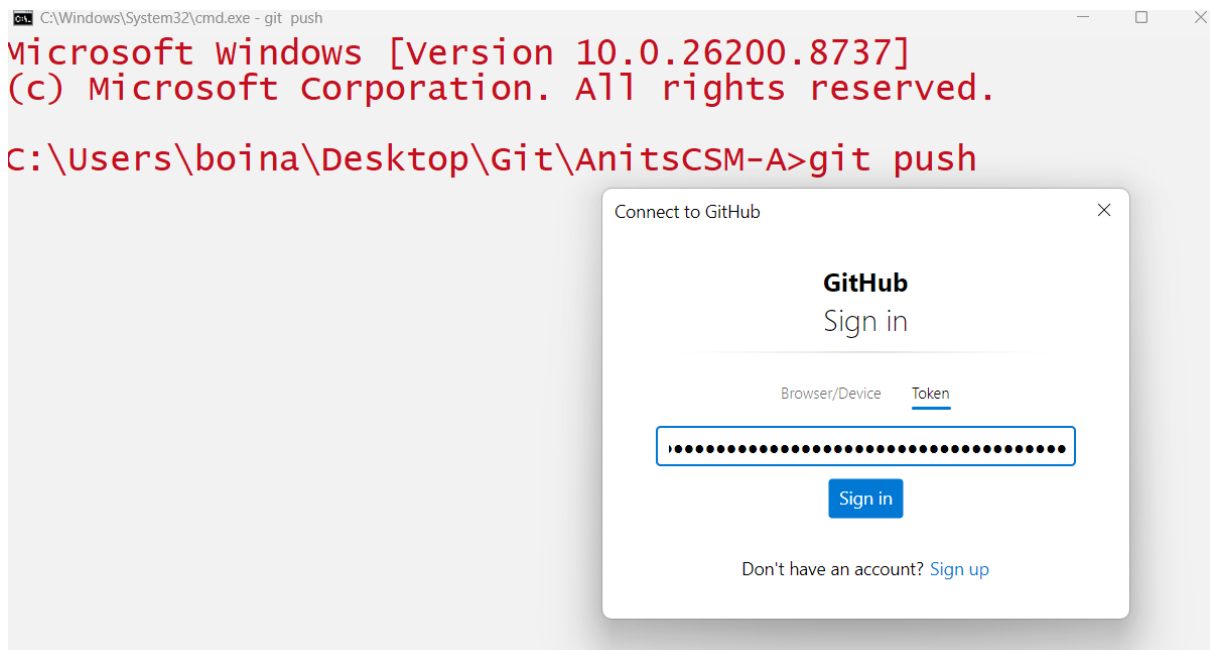
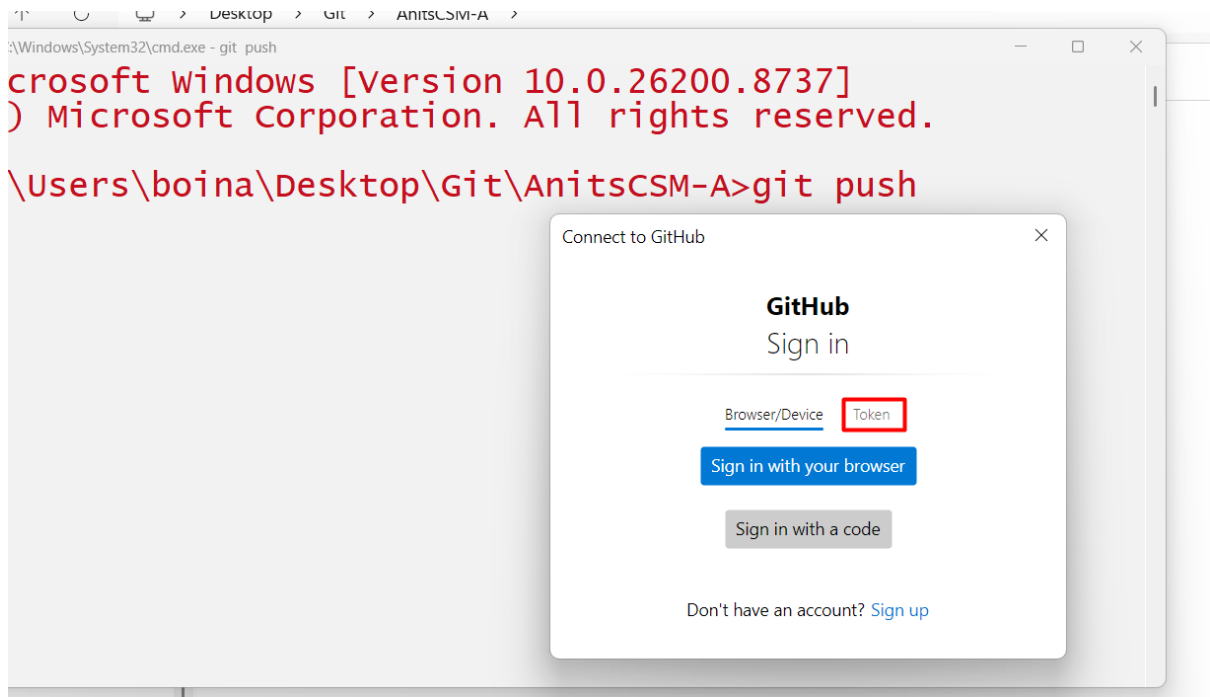
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        style.css

nothing added to commit but untracked files present (use "git add" to track)
C:\Users\boina\Desktop\Git\AnitsCSM-A>_
```

`git push`

In the 3rd stage for connecting to remote repo from local git directory you need to provide token that you generated and saved.

Screenshots for establishing trust between local repository to remote repository.



Now 1st trust will be established then data will be updated in remote repo.

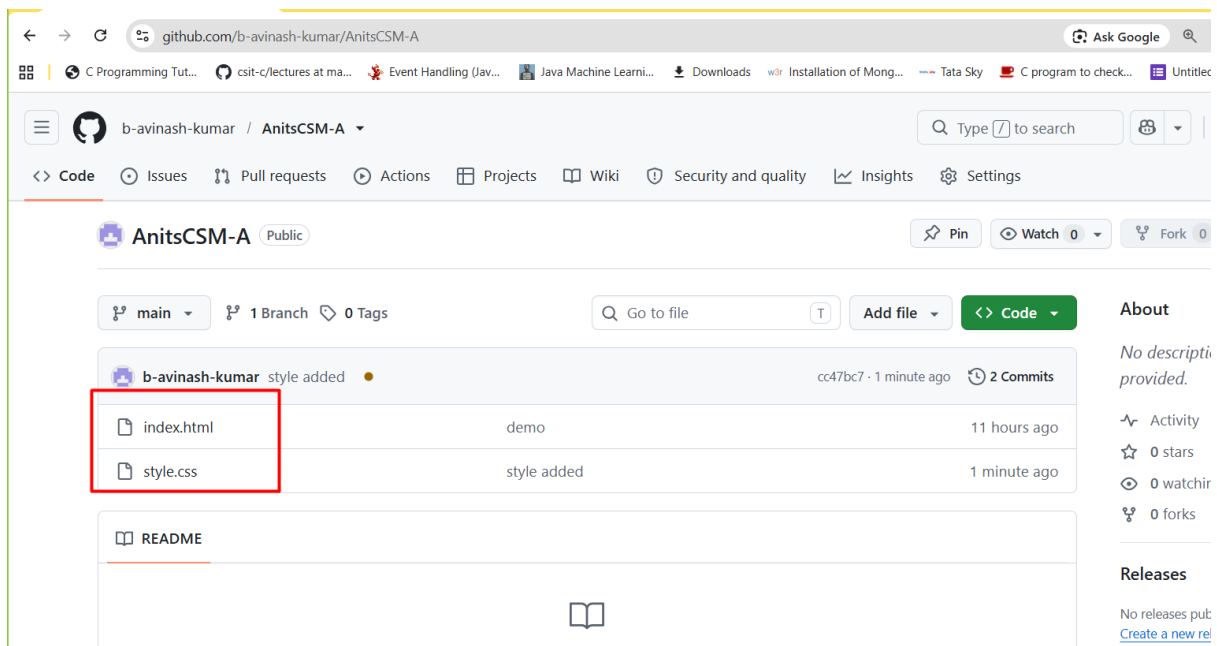
Do once again

```
C:/>git add .
```

```
C:/>git commit -m "added"
```

```
C:/>git push
```

```
C:\Users\boina\Desktop\Git\AnitsCSM-A>git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 254 bytes | 254.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/b-avinash-kumar/AnitsCSM-A.git
7d33292..cc47bc7  main -> main
```

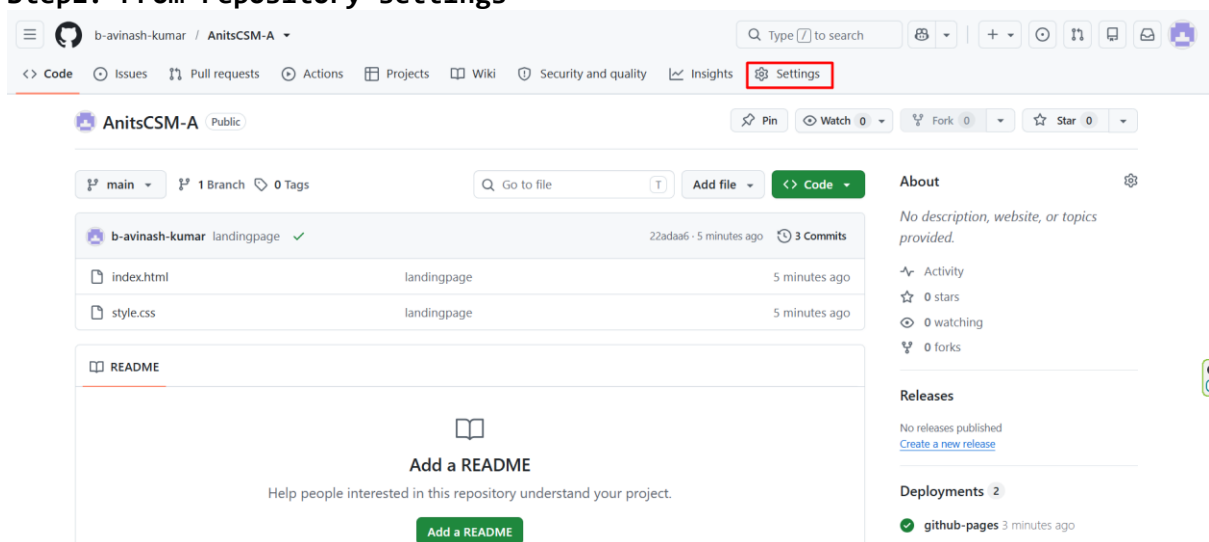


Now lets understand hosting your lab tasks and assignments in github pages.

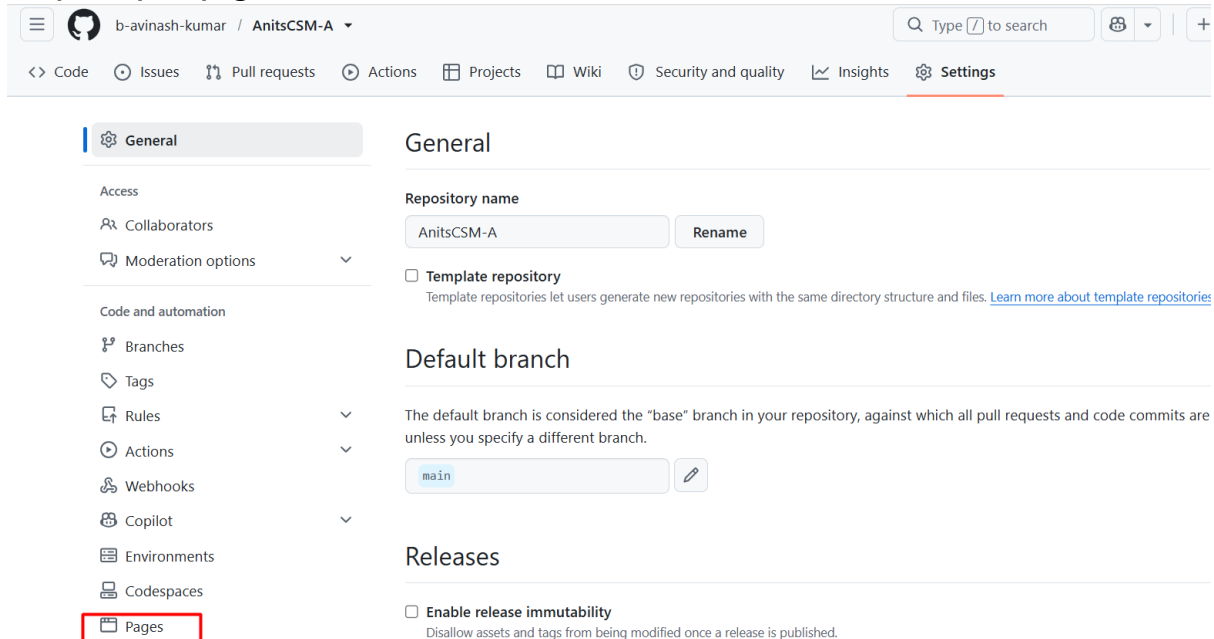
Note: Github pages can help us to host only static web sites.

Configuring our repository(containing web application files) as web site using github pages.

Step1: From repository settings

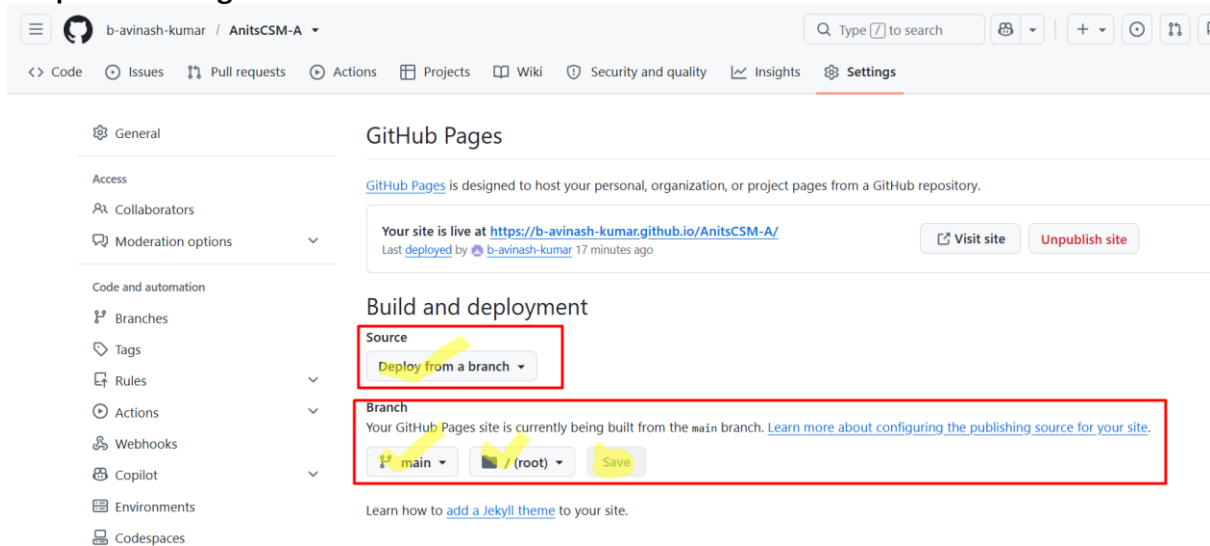


Step2: open pages



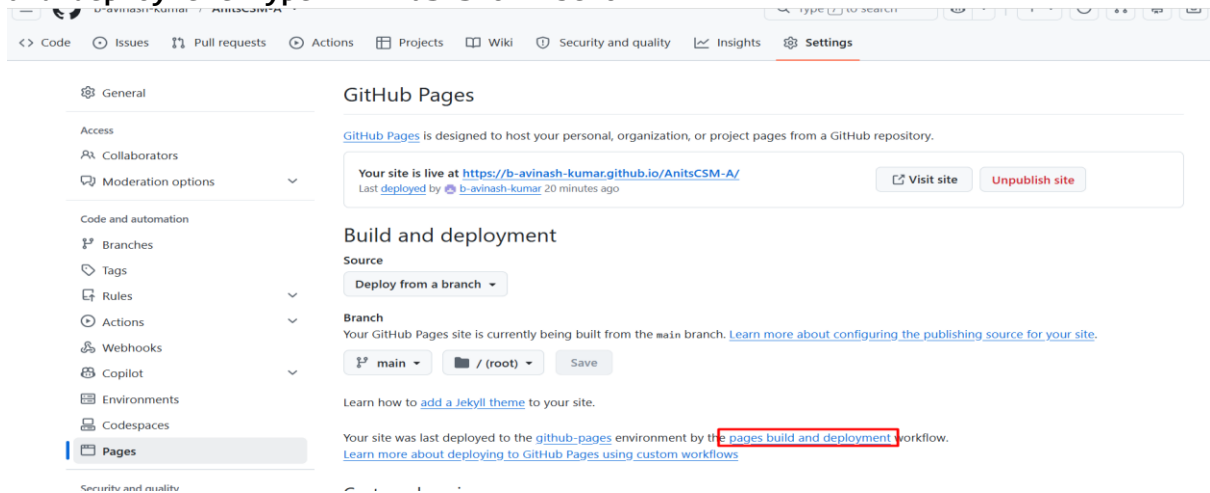
The screenshot shows the GitHub repository settings for 'b-avinash-kumar / AnitsCSM-A'. The 'General' tab is selected. In the left sidebar, the 'Pages' option is highlighted with a red box. The main content area shows the 'General' settings, including the repository name 'AnitsCSM-A', a 'Rename' button, and a checkbox for 'Template repository'. The 'Default branch' is set to 'main'. The 'Releases' section has a checkbox for 'Enable release immutability'.

Step 3: Configurations we need to do



The screenshot shows the GitHub repository settings for 'b-avinash-kumar / AnitsCSM-A' with the 'GitHub Pages' tab selected. The 'Build and deployment' section is highlighted with a red box. It shows the 'Source' is 'Deploy from a branch' and the 'Branch' is 'main'. The 'Branch' section also shows the root directory is '/' and a 'Save' button. The 'Your site is live at' URL is 'https://b-avinash-kumar.github.io/AnitsCSM-A/'.

Step 4: Now after saving, you need to copy your domain from pages build and deployment hyperlink as shown below



The screenshot shows the GitHub repository settings for 'b-avinash-kumar / AnitsCSM-A' with the 'GitHub Pages' tab selected. The 'Build and deployment' section is highlighted with a red box. It shows the 'Source' is 'Deploy from a branch' and the 'Branch' is 'main'. The 'Branch' section also shows the root directory is '/' and a 'Save' button. The 'Your site is live at' URL is 'https://b-avinash-kumar.github.io/AnitsCSM-A/'. The 'Your site was last deployed to the github-pages environment by the pages build and deployment workflow.' is highlighted with a red box.

Step 5: Domain name is the url used to load the page in the web browser

<https://b-avinash-kumar.github.io/AnitsCSM-A/>

The screenshot shows the GitHub Actions interface for the workflow 'pages build and deployment #2'. The workflow is triggered via dynamic 21 minutes ago and has a status of 'Success'. The total duration is 35s and there is 1 artifact. The workflow steps are: build (19s), report-build-status (3s), and deploy (8s). The deploy step is highlighted with a red box and contains the text 'This is your domain' and the URL 'https://b-avinash-kumar.github.io/AnitsCSM-A/'.

pages build and deployment #2

Summary

All jobs

- build
- report-build-status
- deploy

Run details

Usage

Triggered via dynamic 21 minutes ago

Status: Success

Total duration: 35s

Artifacts: 1

pages-build-deployment
on: dynamic

build 19s

report-build-status 3s

deploy 8s

This is your domain

<https://b-avinash-kumar.github.io/AnitsCSM-A/>

Finally Hosted... Congratulations

The screenshot shows a web browser window with the URL 'b-avinash-kumar.github.io/AnitsCSM-A/'. The browser displays a large green and yellow banner with the text 'Congratulations..! you hosted the static web application. Keep doing!!!'.

pages build and deployment - X gradient X +

Ask Gemini

b-avinash-kumar.github.io/AnitsCSM-A/

C Programming Tut... cit-c/lectures at ma... Event Handling (jav... Java Machine Learn... Downloads Installation of Mong... Tata Sky C program to check... Untitled form - Goo... All Bookmarks

Congratulations..! you hosted the static web application. Keep doing!!!

Happy learning