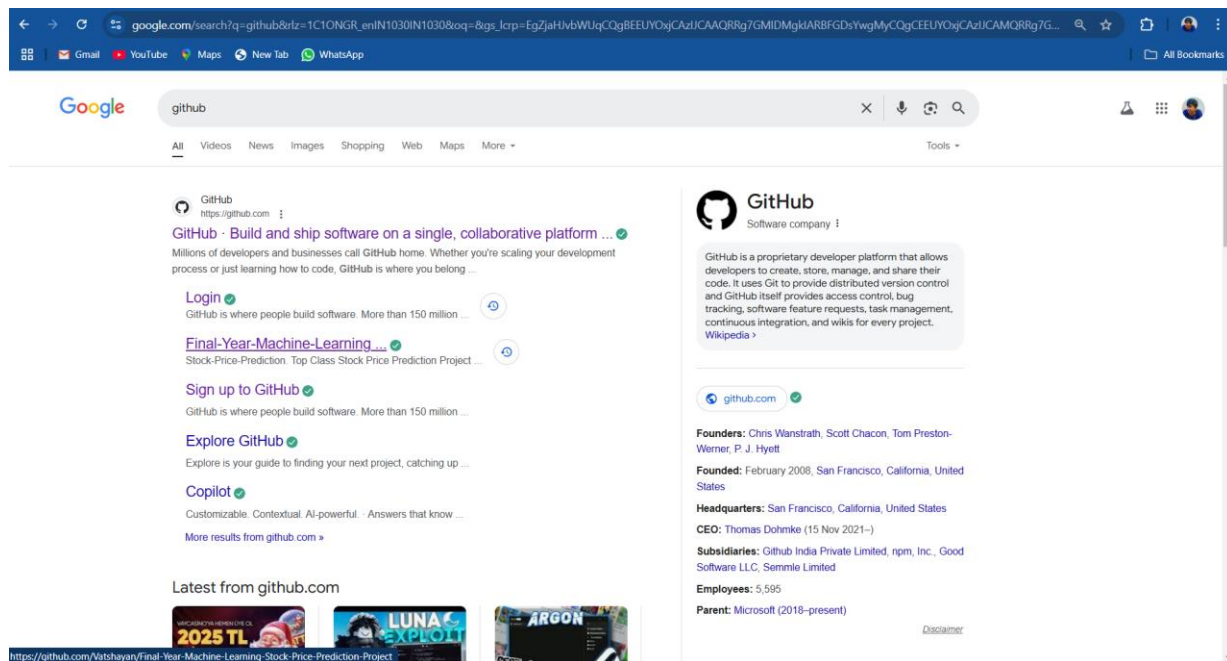


Step-by-Step Guide to Start with GitHub

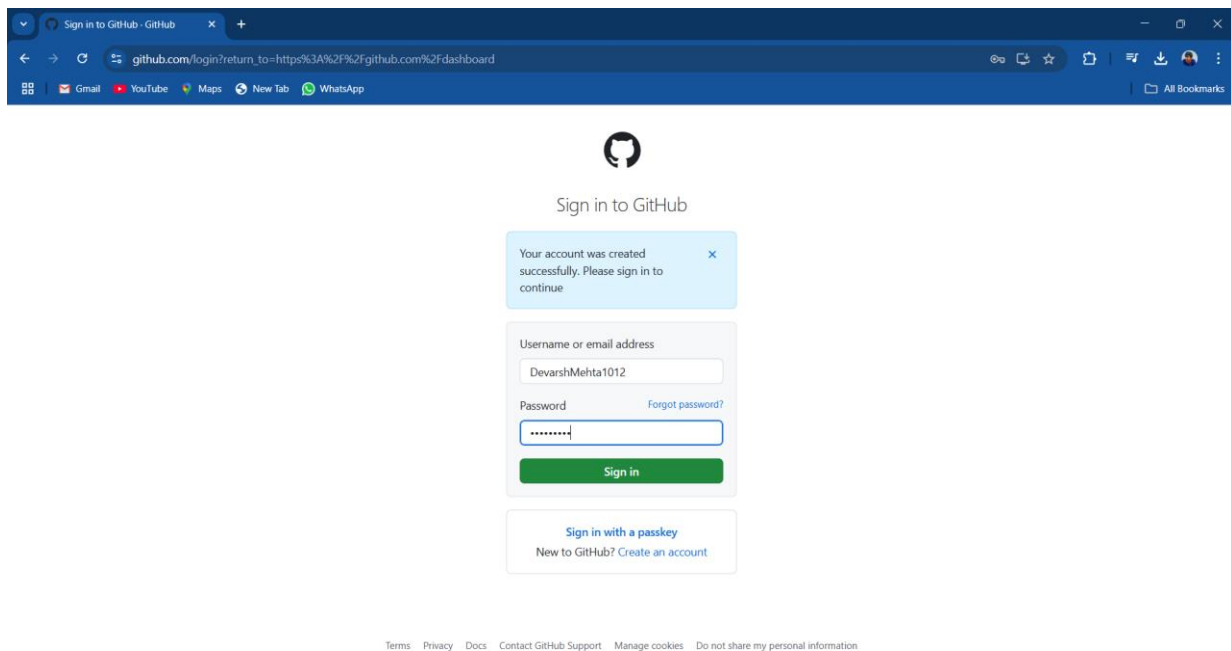
1. Go to GitHub Website:

- Open your browser and navigate to [GitHub](https://github.com).



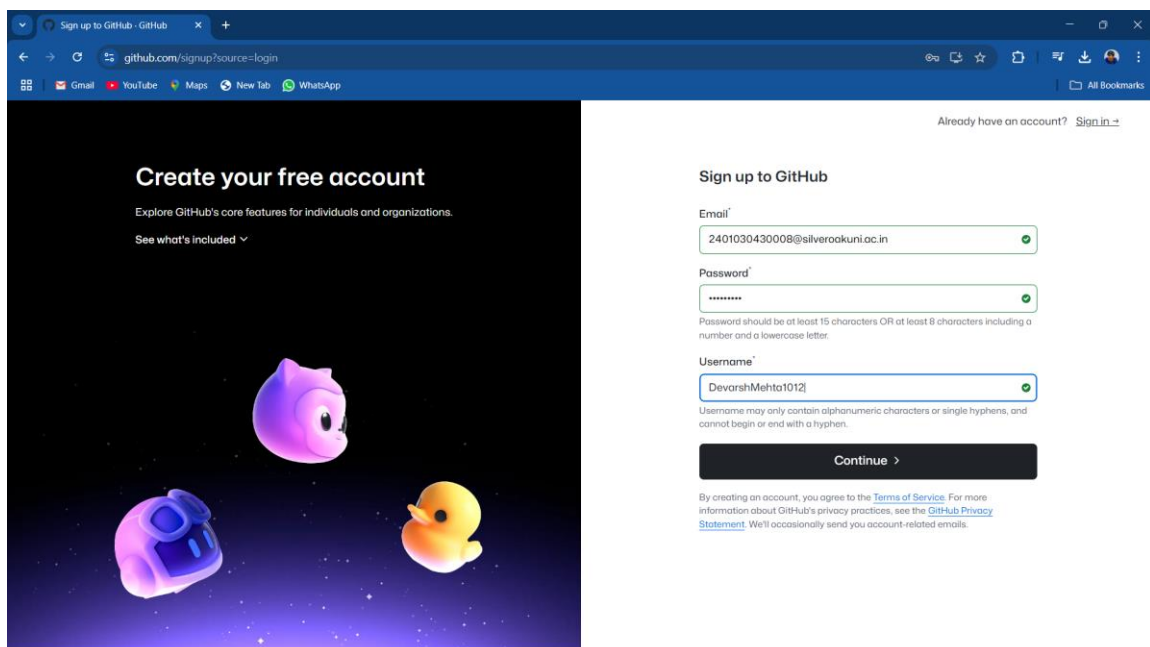
2. Log in to Your GitHub Account:

- If you already have a GitHub account:
- Click on the **"Sign in"** button in the top-right corner.
- Enter your **username or email address** and **password** in the login form.
- Click the **"Sign in"** button to access your account.



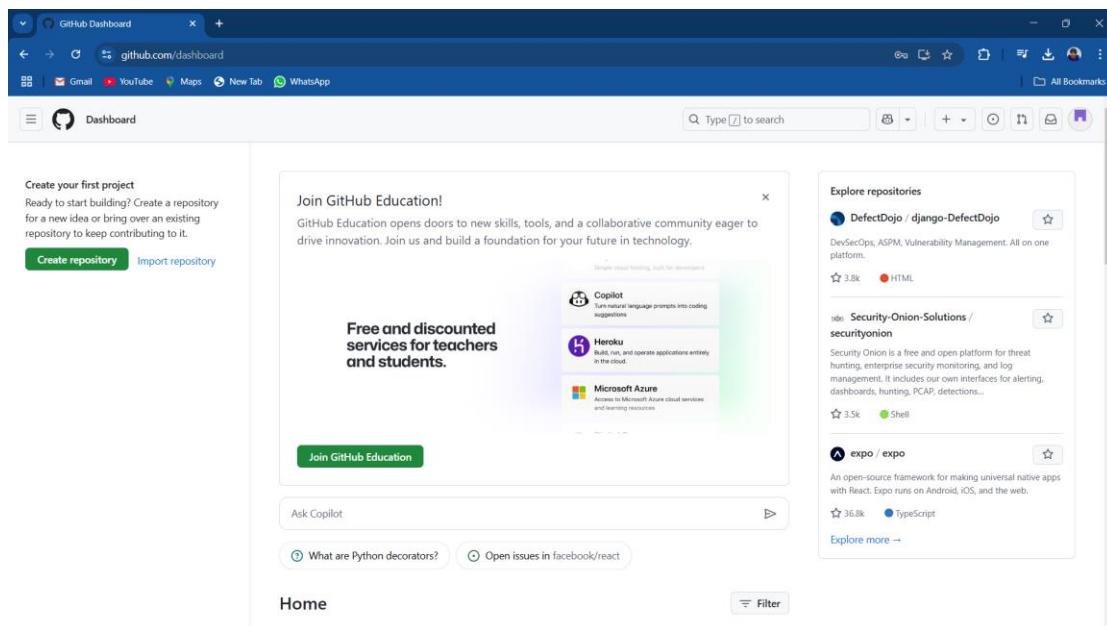
3.If you do not have a GitHub account:

- Click the **"Sign up"** button on the top-right.
- Enter your email address and click **"Continue"**.
- Set a password, username, and verify your email address by following the instructions provided.
- Complete the sign-up process to create a new account.



4. Access Your Dashboard

- After logging in, you'll be redirected to your **GitHub Dashboard**.
- This is your home page, where you can see all your repositories, recent activity, and shortcuts for creating new repositories.



5. Create a New Repository:

- From the dashboard, click the green **"New"** button on the left panel or navigate to the **Repositories** tab and select **"New"**.
- Fill out the repository details:
- **Repository Name:** Enter your folder name (**PRJ**).
- **Privacy:** Choose between:
 - **Public** (anyone can see it).

github.com/new

New repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

DevarshMehta1012 /

Great repository names are short and memorable. Need inspiration? How about [didactic-guacamole](#) ?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

6.Create the Repository:

- Click the "Create repository" button at the bottom of the page.

New repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

DevarshMehta1012 / 2401030430008

2401030430008 is available.

Great repository names are short and memorable. Need inspiration? How about [didactic-guacamole](#) ?

Description (optional)

7.Steps for adding a folder:

1.Initialize a New Repository

```
devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1
$ git init
Initialized empty Git repository in C:/Users/devarsh mehta/OneDrive/AppData/Desktop/Repository/Assignment_1/.git/
devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (master)
$ |
```

2. Add the File

```
devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (master)
$ git add .

devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (master)
$ |
```

3. Check the status of

```
devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (master)
$ git add .

devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   clg project (2).pdf
        new file:   git documentation .rtf
```

4.Commit the Changes with a Message

```
devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (master)
$ git commit -m "First commit"
[master (root-commit) e671c25] First commit
 2 files changed, 814661 insertions(+)
 create mode 100644 clg project (2).pdf
 create mode 100644 git documentation .rtf

devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (master)
$
```

5.Rename the Branch to master

```
devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (master)
$ git branch -M main

devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (main)
$
```

6.Link the Local Repository to a Remote GitHub Repository

```
devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (master)
$ git remote add origin-pish https://github.com/DevarshMehta1012/2401030430008.git

devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/Repository/Assignment_1 (master)
$ |
```

7.Push the Local Changes to GitHub

```
devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/project/Assignment_1 (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 39.97 MiB | 655.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: warning: See https://gh.io/lfs for more information.
remote: warning: File clg project (2).pdf is 53.88 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/DevarshMehta1012/2401030430008/pull/new/master
remote:
To https://github.com/DevarshMehta1012/2401030430008.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

devarsh mehta@Devarsh MINGW64 ~/OneDrive/AppData/Desktop/project/Assignment_1 (master)
$ |
```

8. Folder are add in repository:

The screenshot shows the GitHub interface for a repository named '2401030430008' by user 'DevarshMehta1012'. The repository is public and has 0 stars, 1 watching, and 0 forks. The main branch is 'main', which had recent pushes 47 minutes ago. The repository contains two files: 'clg project (2).pdf' and 'git documentation .rtf', both committed 36 minutes ago. A 'README' section is visible at the bottom, prompting the user to 'Add a README' to help people understand the project. The right sidebar shows sections for 'About', 'Releases', and 'Packages', all of which are currently empty or have no published content.