Git Hub Registration

Steps1:

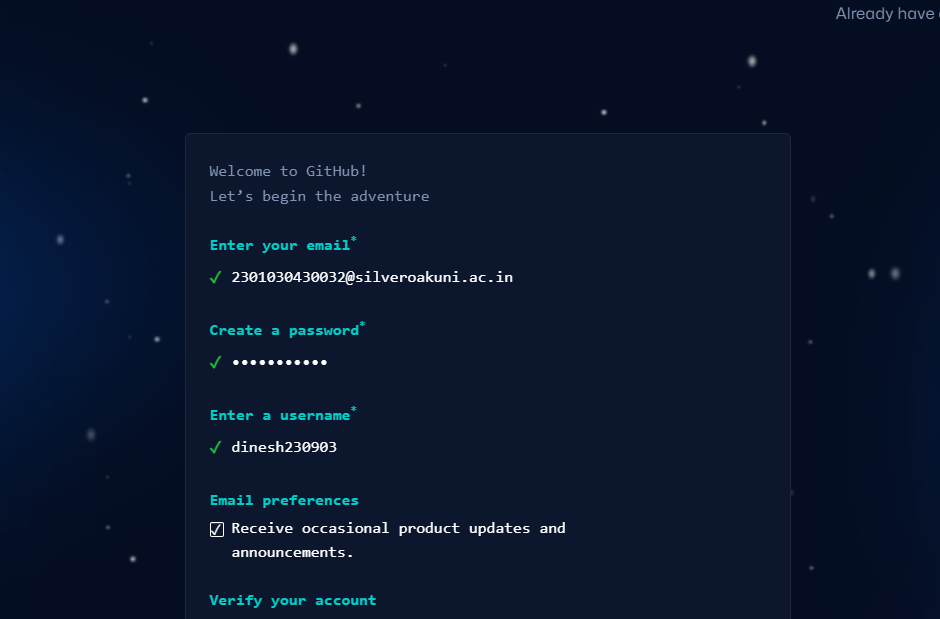
The first steps in starting with GitHub are to create an account, choose a product that fits your needs best, verify your email, set up two-factor authentication, and view your profile.

There are several types of accounts on GitHub. Every person who uses GitHub has their own personal account, which can be part of multiple organizations and teams. Your personal account is your identity on GitHub.com and represents you as an individual.

1. Creating an account

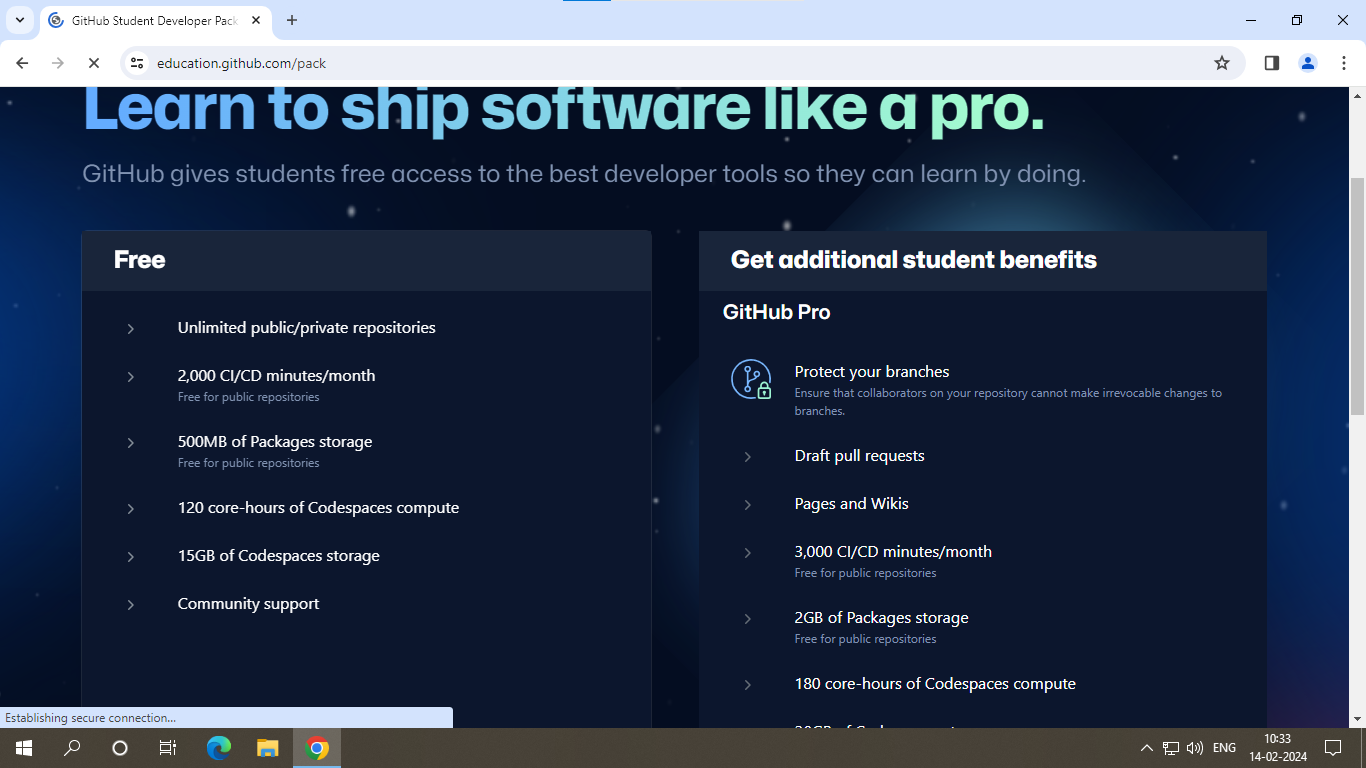
To sign up for an account on GitHub.com, navigate to https://github.com/ and follow the prompts.

To keep your GitHub account secure you should use a strong and unique password. For more information, see "Creating a strong password."



1. Choosing your GitHub product

You can choose GitHub Free or GitHub Pro to get access to different features for your personal account. You can upgrade at any time if you are unsure at first which product you want.



For more information on all of GitHub's plans, see "GitHub’s plans."

1. Verifying your email address

To ensure you can use all the features in your GitHub plan, verify your email address after signing up for a new account. For more information, see "Verifying your email address."

1. Configuring two-factor authentication

Two-factor authentication, or 2FA, is an extra layer of security used when logging into websites or apps. We strongly urge you to configure 2FA for the safety of your account. For more information, see "About two-factor authentication."

Optionally, after you have configured 2FA, add a passkey to your account to enable a secure, passwordless login. For more information, see "About passkeys" and "Managing your passkeys."

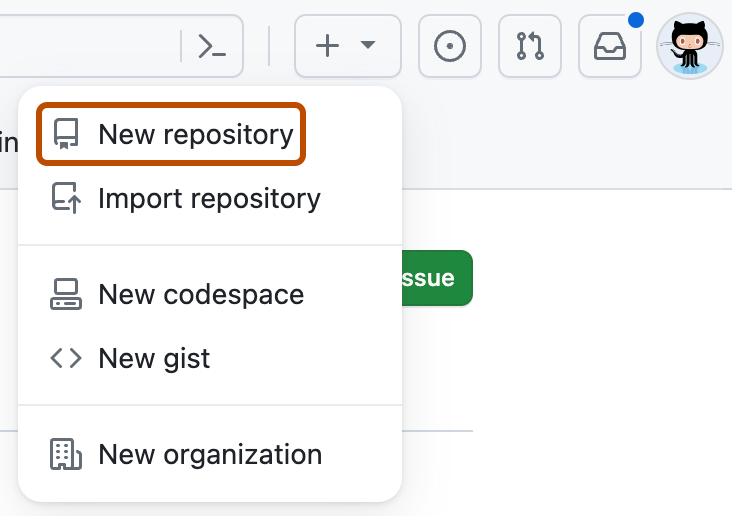
1. Viewing your GitHub profile and contribution graph

Your GitHub profile tells people the story of your work through the repositories and gists you've pinned, the organization memberships you've chosen to publicize, the contributions you've made, and the projects you've created. For more information, see "About your profile" and "Viewing contributions on your profile."

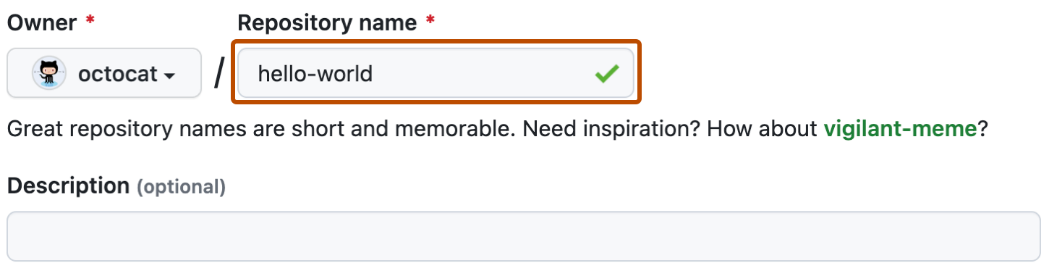
Step2:

1. Create a repository

* In the upper-right corner of any page, select +, then click New repository.



* Type a short, memorable name for your repository. For example, "hello-world".



* Optionally, add a description of your repository. For example, "My first repository on GitHub."
* Choose a repository visibility. For more information, see "About repositories."
* Select Initialize this repository with a README.
* Click Create repository.

1. Open terminal in your code folder:

* Open a terminal window.
* Navigate to your code folder using the cd command. For example, if your code folder is on
* your desktop named "assignment," use cd Desktop/assignment.



1. Initalize a Git repository:

* Run the command git init to create a new Git repository in your code folder.



1. Add your files to staging:

* Use the command git add . to add all files in your current directory (your code folder) to the



* staging area. This tells Git that you want to track these files for version control.

1. Commit your changes:

* Run the command git commit -m "Your commit message". Replace "Your commit message"



* with a concise descripton of the changes you made. This creates a snapshot of your code
* with a message. Ex. “Added Assignment Folders and Documentation.”

1. Add the remote repository :

* Run the command git remote add origin <repository\_URL>, replacing <repository\_URL> with the HTTPS clone URL you copied from GitHub in step 1. This tells Git about the remote repository where you want to push your code.



1. Push your changes to GitHub:

* Run the command git push -u origin <branch\_name>.
* Replace <branch\_name> with the name of the branch you want to push. This is our first push so we’ll use default branch name “master”
* The -u flag sets the remote branch as "upstream" for the current branch, simplifying future pushes.



1. Verify:

* Go to your GitHub repository and refresh the page. You should see your code pushed and
* reflected in the repository

