

Module 2: Working with Git Repositories

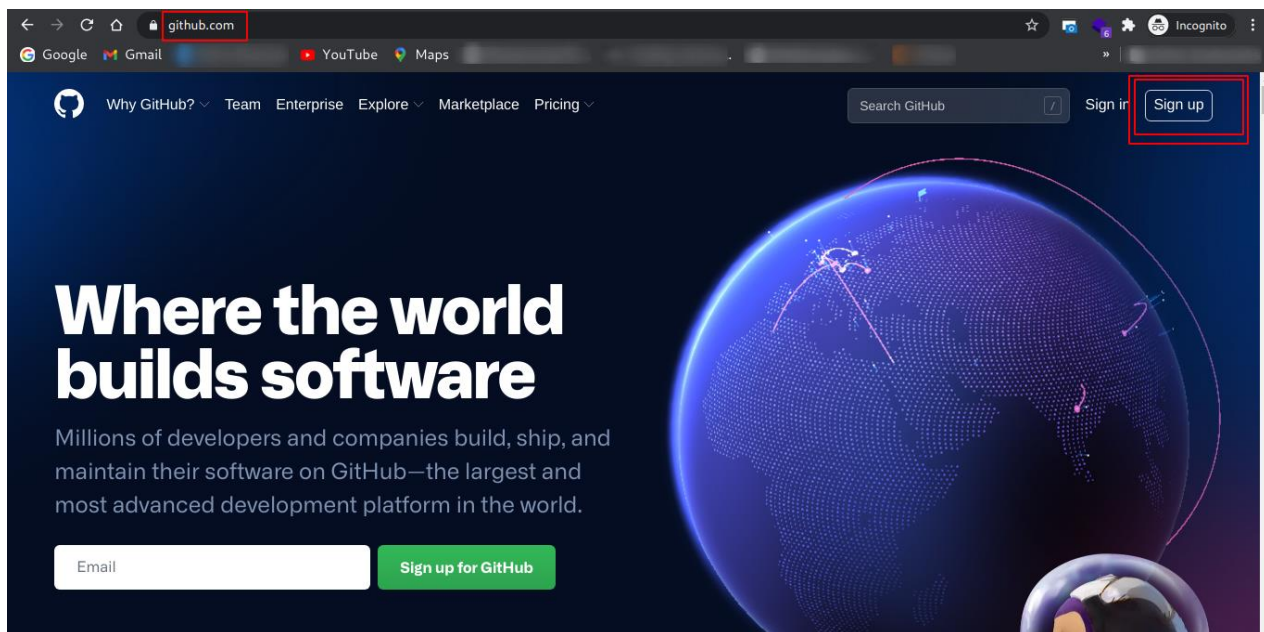
Demo 1: Demo on GitHub Setup

Problem Statement:

How we work on GitHub.

Solution:

Step 1: In this demo, we are creating our account on GitHub. For that you must go on <https://www.github.com/> . After that click on the **signup** button.



Step 2: Create account form appeared, enter your username (which you want to use as your GitHub username), enter your email address (which you want to use as GitHub account) and then enter your password (it should be unique and complex), and after all this click on verify your account.

Join GitHub

Create your account

Username *
demousername123 ✓

Email address *
[redacted] ✓

Password *
[redacted] ✓

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Email preferences
☒ Send me occasional product updates, announcements, and offers.

Verify your account

Step 3: After completing all the steps click on “create account”.

Join GitHub

Create your account

Username *
demousername123 ✓


Email address *
[redacted] ✓

Password *
[redacted] ✓

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Email preferences
☒ Send me occasional product updates, announcements, and offers.

Verify your account



Create account

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

Step 4: After creating your account, you got a welcome page in which you got options “**What kind of work do you do, mainly?**”, “**How much programming experience do you have?**”, “**What do**

you plan to use GitHub for?” and **“I am interested in:”** options, after completing this answer click on “next step”.

Welcome to GitHub

Woohoo! You've joined millions of developers who are doing their best work on GitHub.

Tell us what you're interested in. We'll help you get there.

What kind of work do you do, mainly?

Software Engineer I write code	Student I go to school
Product Manager I write specs	UX & Design I draw interfaces
Data & Analytics I write queries	Marketing & Sales I look at charts
Teacher I educate people	Other I do my own thing

How much programming experience do you have?

None I don't program at all	A little I'm new to programming
A moderate amount I'm somewhat experienced	A lot I'm very experienced

Step 5: In this step you must verify your email address by opening and verify the verification email sent by GitHub to you.



Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to [redacted].

[Resend verification email](#)

[Change your email settings](#)

Check your email, open GitHub mail, and then click on “verify email address”.



Almost done, @demousername123!

To complete your GitHub sign up, we just need to verify your email address:

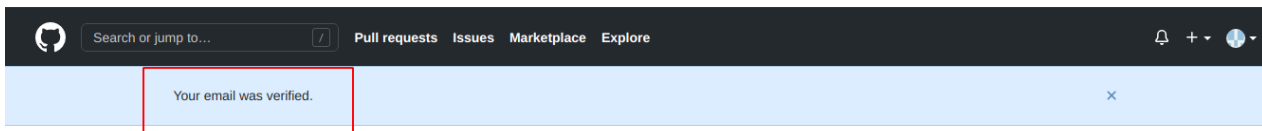
[Verify email address](#)

Once verified, you can start using all of GitHub's features to explore, build, and share projects.

Button not working? Paste the following link into your browser:

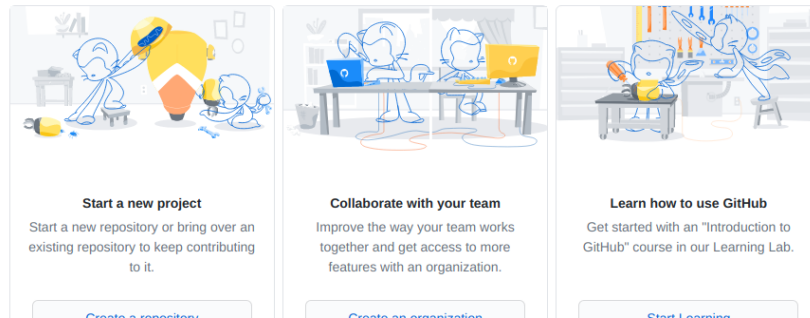
https://github.com/users/demousername123/emails/159550052/confirm_verification

verification is completed as you can see on your GitHub dashboard.



What do you want to do first?

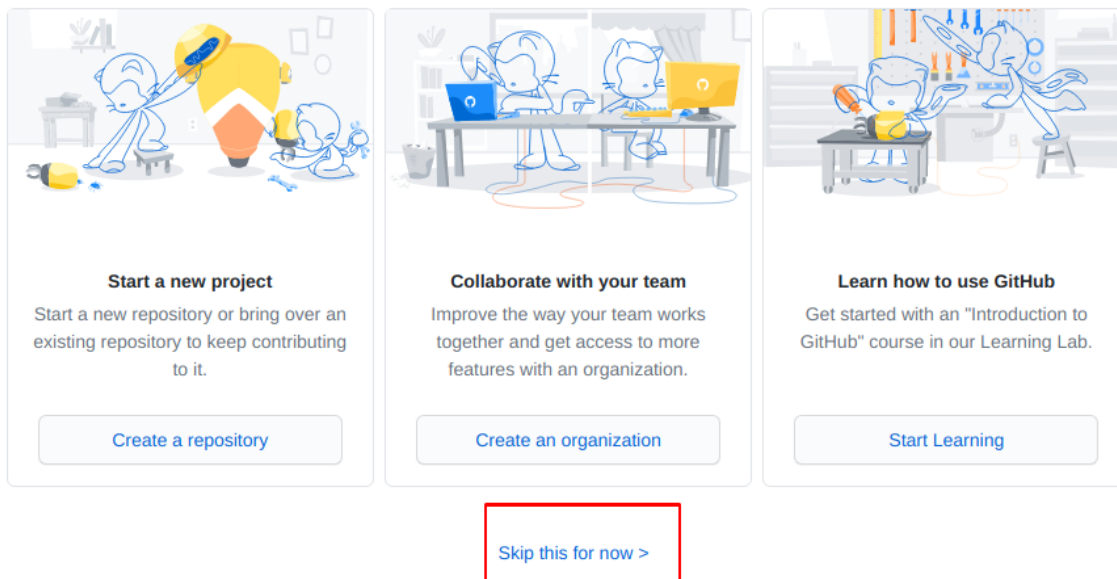
Every developer needs to configure their environment, so let's get your GitHub experience optimized for you.



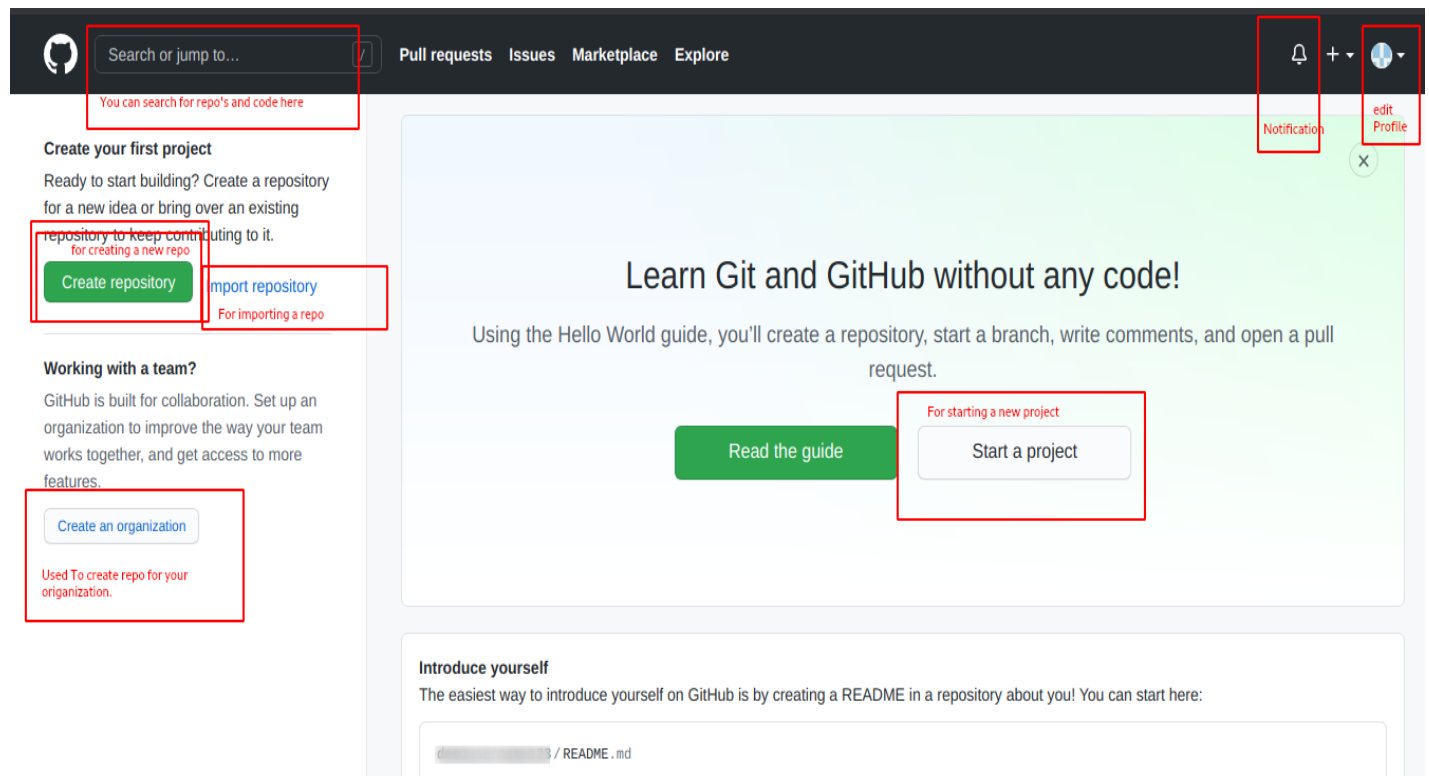
Step 6: Now you can see here there are several options, like **“Start a new project”**, **“Collaborate with your team”**, **“Learn how to use GitHub”**, By scrolling down you get an option of **“Skip this for now”** click onto it.

What do you want to do first?

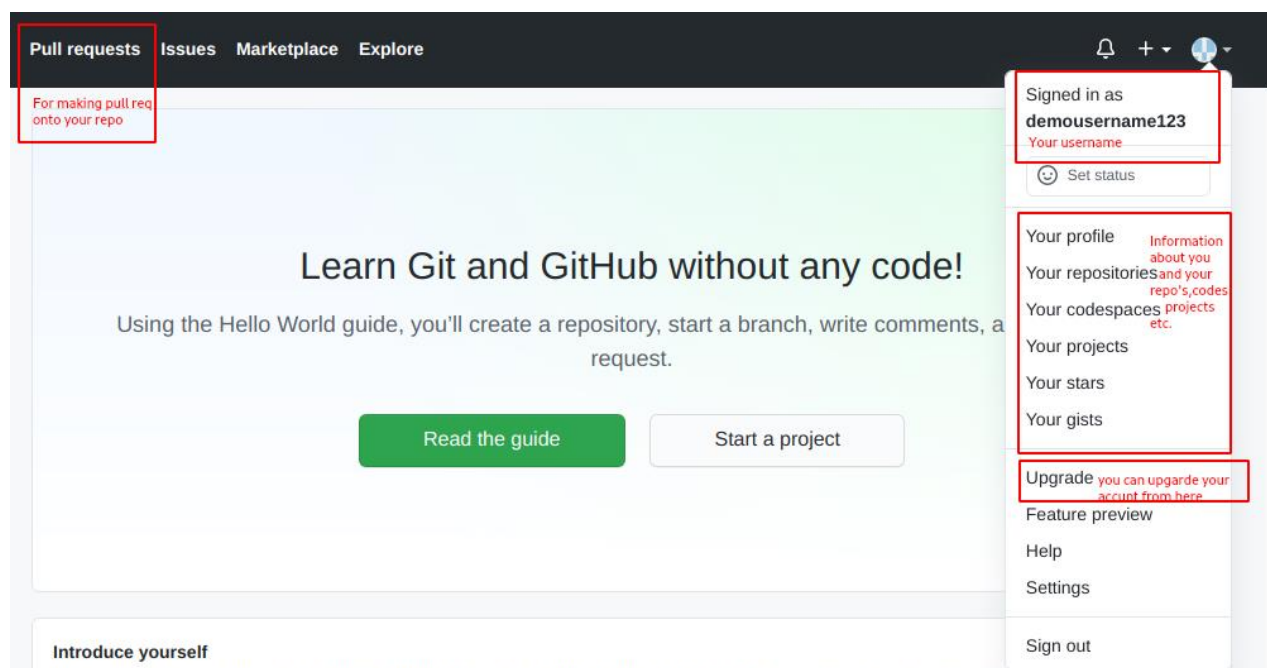
Every developer needs to configure their environment, so let's get your GitHub experience optimized for you.



after clicking onto this, you got your GitHub dashboard in which you get several options there, let us have a look into this.



on the right side we have, Several other options like,



Step 7 : Let's create our first Project click on **"Start a Project"**, after that you got onto a web page which will ask you for your **"Repository Name"** , **"Description"** and there are two options also


Public and **Private** to make your repository public or private(public is accessible by all but private is only you can see it), There are other options also , like **add a README File** , **add .gitignore** and **choose a license** .

After completing these steps click on **“Create Repository”**, and your first repository created here.

Create a new repository


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


Owner * **Repository name ***



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

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:
Skip this step if you're importing an existing repository.

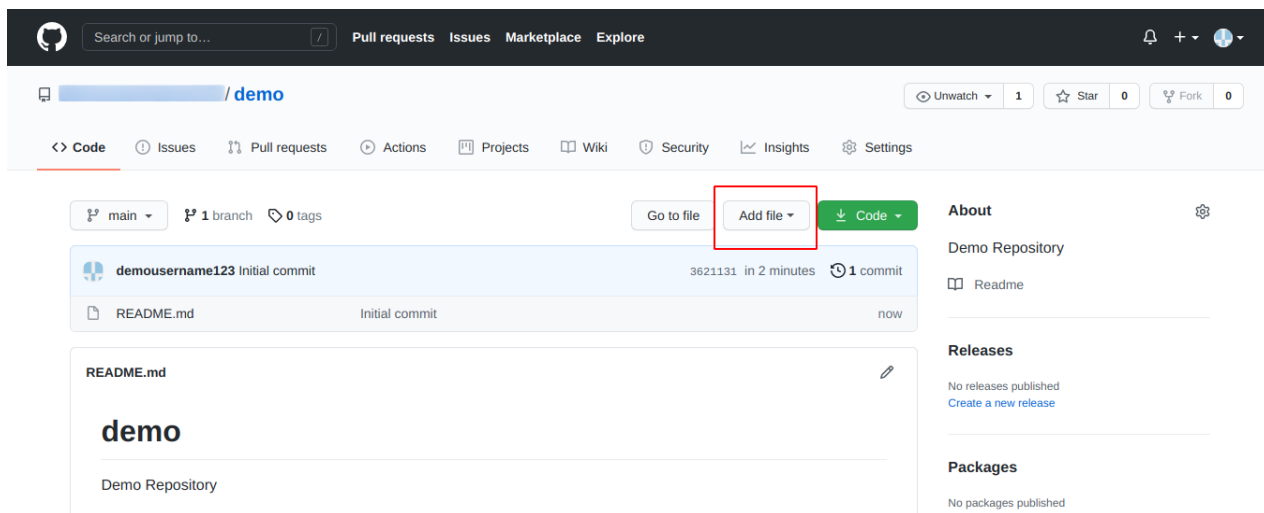
☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

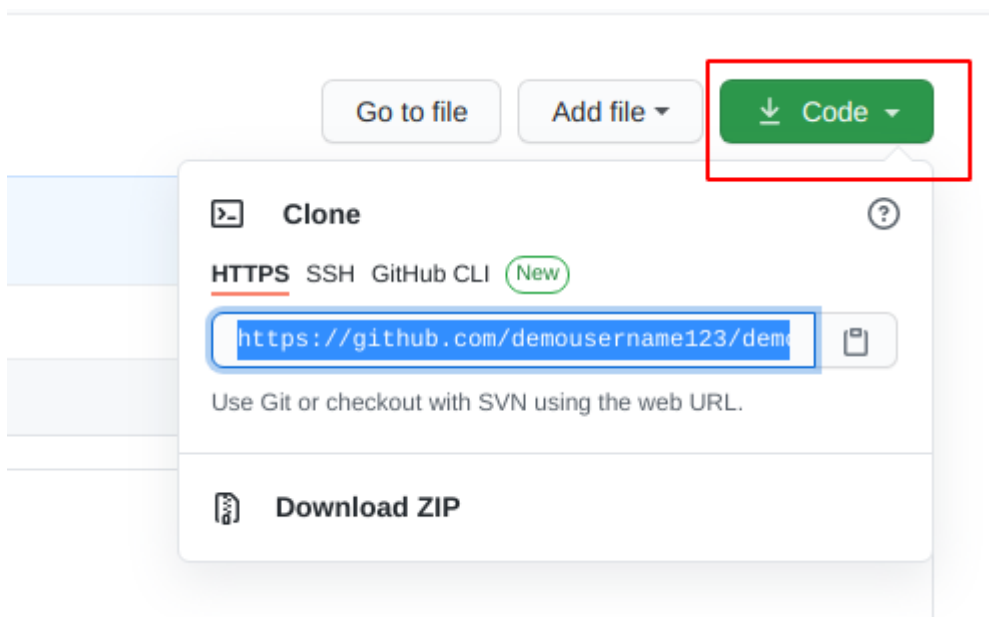
☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

After clicking on **“creating repository”**, You can see you got your repository created here, you can add file here by clicking on **“Add File”**.

Module 2 – Working with Git Repositories

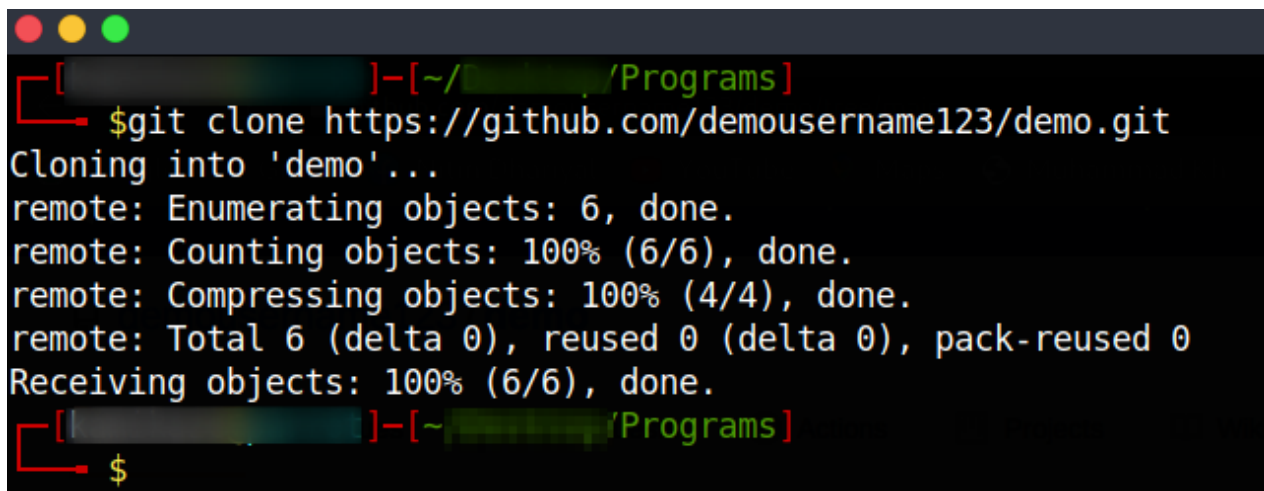


Step 8: I committed a file into my git repository. Let us get this file into our local system, using the *git clone* command. You get the link of that repository by clicking on “code”. Here you got a link for your GitHub repository which you want to clone into your local system.



After, getting this link open your Git Bash onto that folder in which you want to clone this file. and use command:

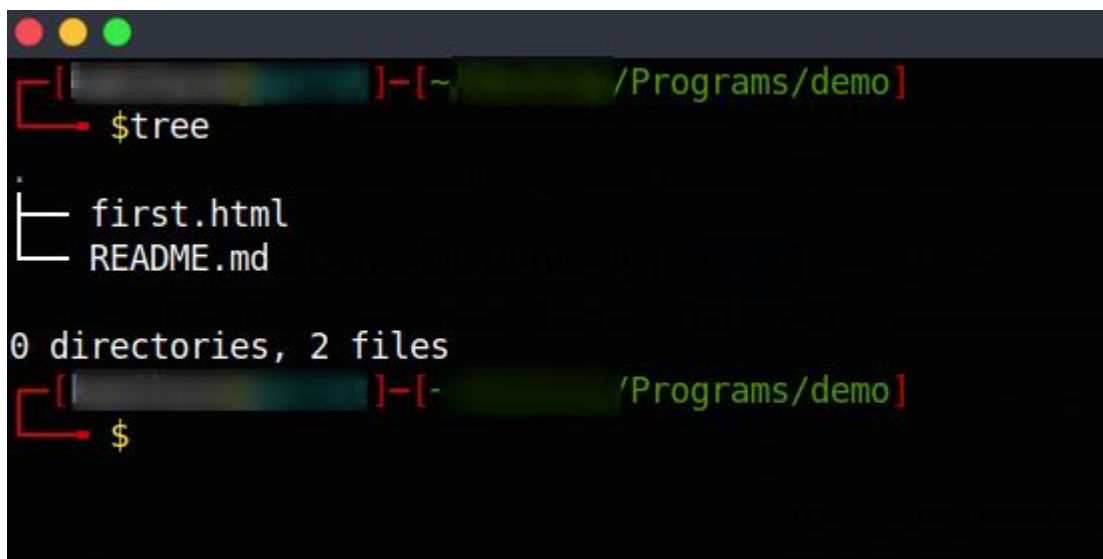
Command Used: `git clone link_of_git_repo_you_want_to_clone`

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The prompt is [redacted]~[redacted]/Programs. The command \$git clone https://github.com/demousername123/demo.git is entered. The output shows the cloning process: Cloning into 'demo'..., remote: Enumerating objects: 6, done., remote: Counting objects: 100% (6/6), done., remote: Compressing objects: 100% (4/4), done., remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0, and Receiving objects: 100% (6/6), done. The prompt returns to [redacted]~[redacted]/Programs \$.

```
[redacted]~[redacted]/Programs]
$git clone https://github.com/demousername123/demo.git
Cloning into 'demo'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
[redacted]~[redacted]/Programs]
$
```

Step 9: Let us check whether the repository got cloned or not you can use some of the basic commands for listing like **ls** in Linux and **dir** in windows. You can also use the **tree** command here if you are using bash.

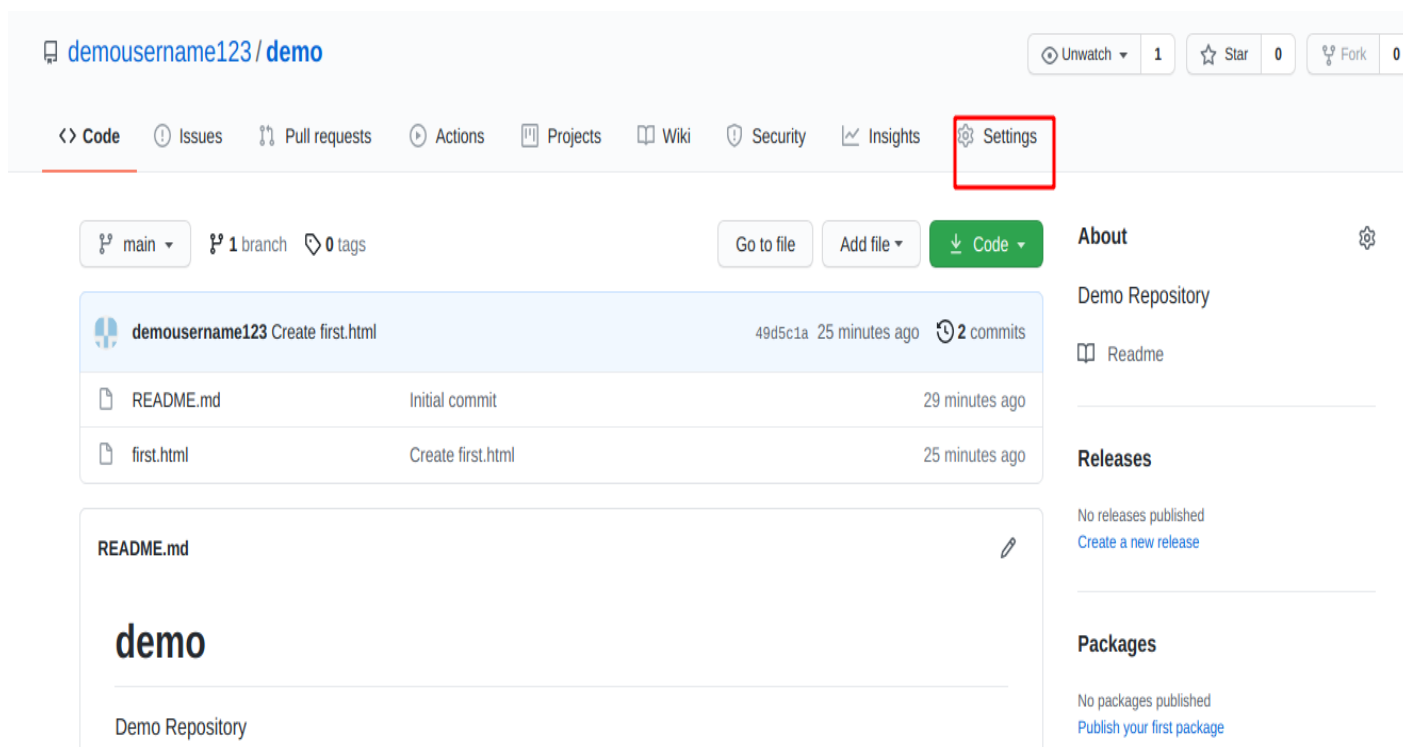
command used: **ls** (in Linux), **dir** (in windows)

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The prompt is [redacted]~[redacted]/Programs/demo. The command \$tree is entered. The output shows a directory tree with first.html and README.md. Below the files, it says 0 directories, 2 files. The prompt returns to [redacted]~[redacted]/Programs/demo \$.

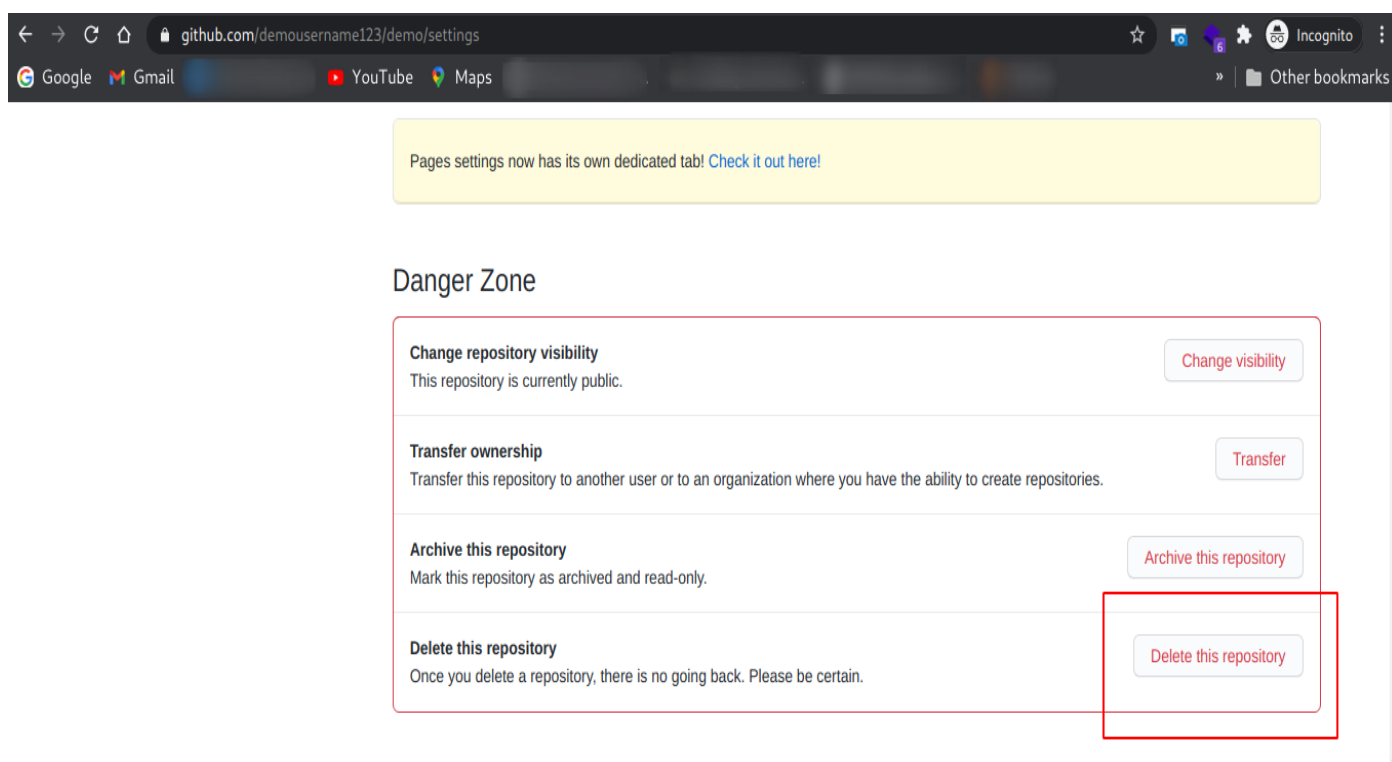
```
[redacted]~[redacted]/Programs/demo]
$tree
.
├── first.html
└── README.md

0 directories, 2 files
[redacted]~[redacted]/Programs/demo]
$
```

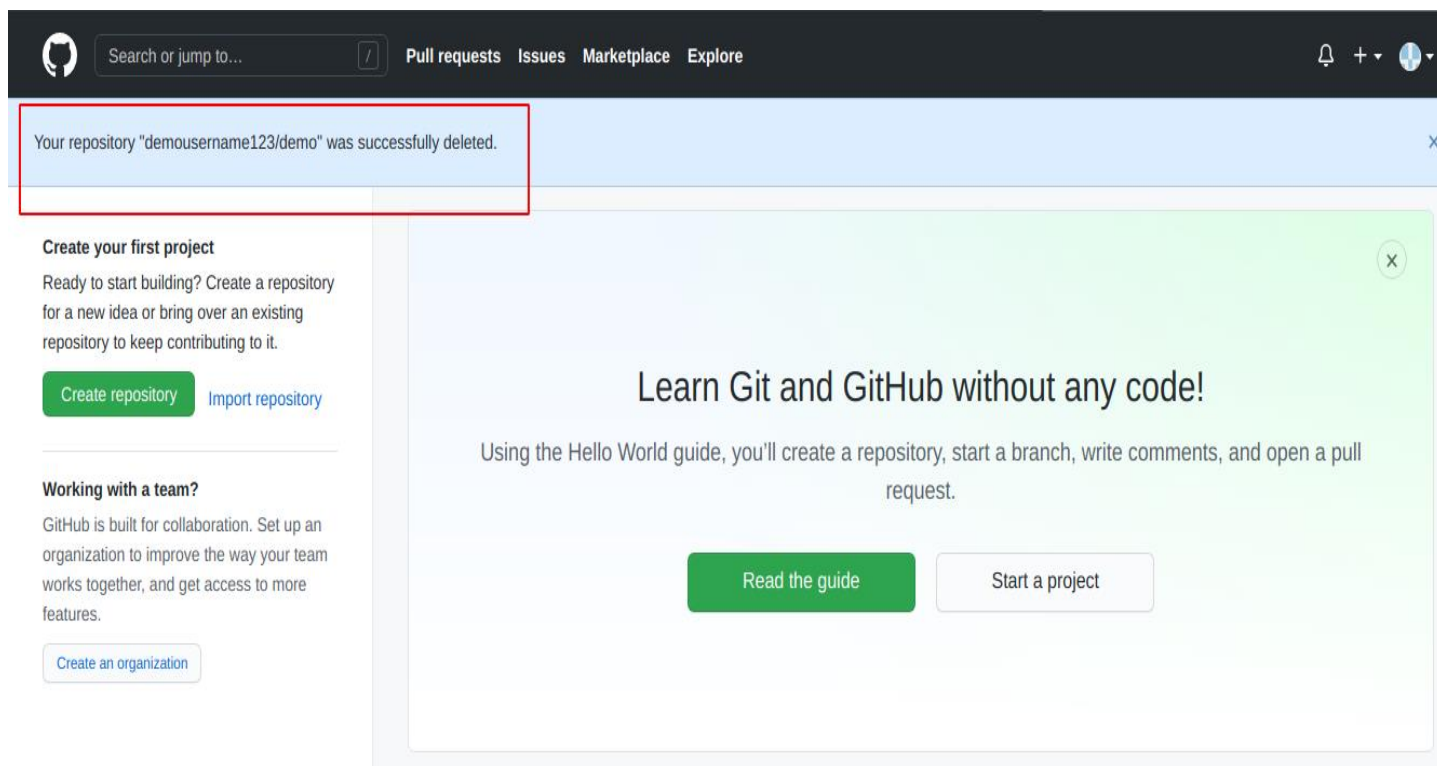
Step 10: Let us delete this repository, for deleting any repository click on “**settings**” option.



Step 11: After clicking on “**Settings**”, Go through the page and scroll down to the page, you got an option of “**Delete this repository**”.



Step 12: After completing the last step you will see that your repository gets deleted.



The screenshot shows the GitHub homepage. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. A light blue notification banner at the top left states: "Your repository 'demousername123/demo' was successfully deleted." The left sidebar contains two sections: "Create your first project" with buttons for "Create repository" and "Import repository", and "Working with a team?" with a button for "Create an organization". The main content area features a large green box with the heading "Learn Git and GitHub without any code!" and the text "Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request." Below this text are two buttons: "Read the guide" and "Start a project".

Search or jump to... Pull requests Issues Marketplace Explore

Your repository "demousername123/demo" was successfully deleted.

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repository Import repository

Working with a team?

GitHub is built for collaboration. Set up an organization to improve the way your team works together, and get access to more features.

Create an organization

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide Start a project