

Lesson 02 Demo 01

Creating and Cloning a GitHub Repository

Objective: To create and clone a GitHub repository for understanding the fundamentals of repository management and version control using Git and GitHub

Tools required: Git and GitHub

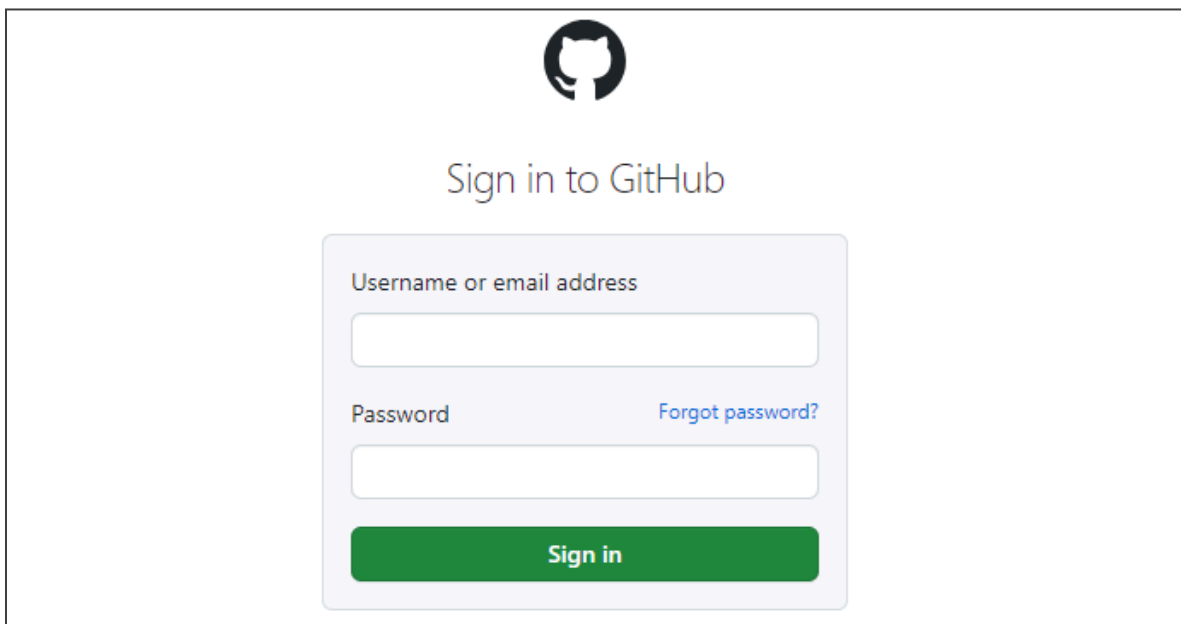
Prerequisites: You need to have Git installed and a GitHub account to proceed with this demo.

Steps to be followed:

1. Create a new GitHub repository
2. Clone the GitHub repository

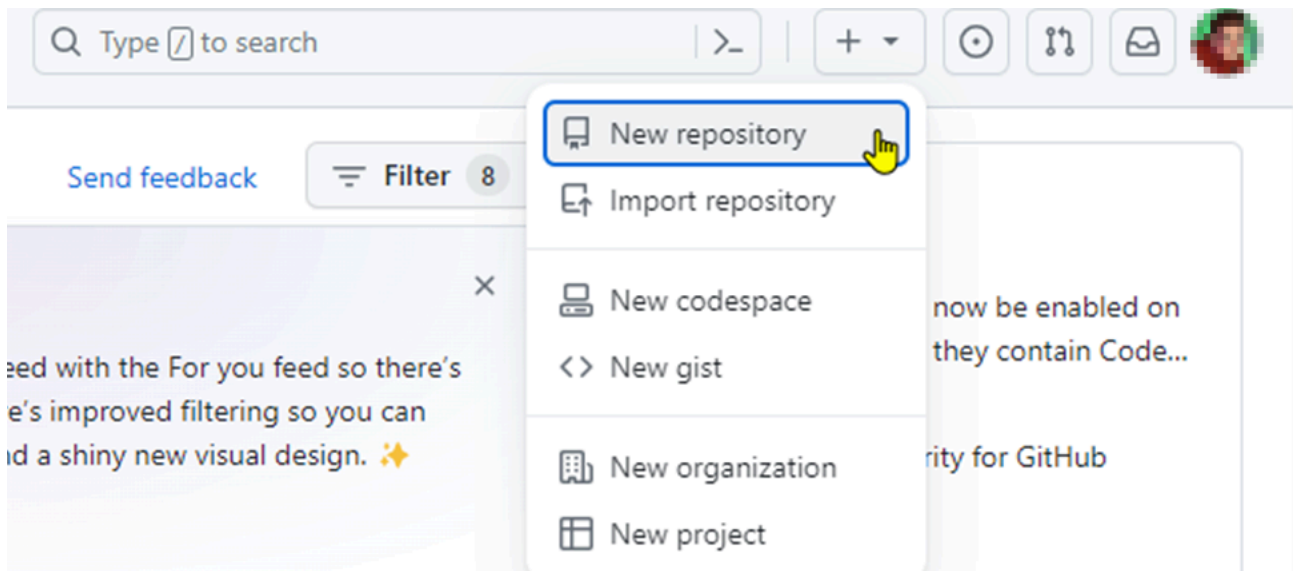
Step 1: Create a new GitHub repository

1.1 Open the browser in your lab, go to **github.com**, and log in to your account

A screenshot of the GitHub sign-in page. At the top center is the GitHub logo (an octocat). Below it, the text "Sign in to GitHub" is displayed. Underneath is a light gray rounded rectangle containing the login form. The form has two input fields: the first is labeled "Username or email address" and the second is labeled "Password". To the right of the password field is a blue link that says "Forgot password?". At the bottom of the form is a green button with the text "Sign in" in white.

Note: If you do not have a GitHub account, visit the official website at <https://github.com/signup> and create a new account

- 1.2 Click on the + icon from the upper-right corner of the page and select **New repository** from the drop-down menu





- 1.3 Enter the **Repository name** and **Description** as shown below:

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 *

✔ Your new repository will be created as lesson-end-project.
The repository name can only contain ASCII letters, digits, and the characters `.`, `-`, and `_`.

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

Description (optional)

Note: The repository name and description may differ from those shown in the screenshot.


1.4 Choose **Public** for the repository type


Owner * / Repository name *
lesson-end project

✓ Your new repository will be created as lesson-end-project.
The repository name can only contain ASCII letters, digits, and the characters -, ., and _.


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


Description (optional)
This is the lesson-end project for this lesson.

☒  **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.

1.5 Select **Add a README file** option to include a README file for the repository

☒  **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
License: None ▾

1.6 Click on the **Create repository** button

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

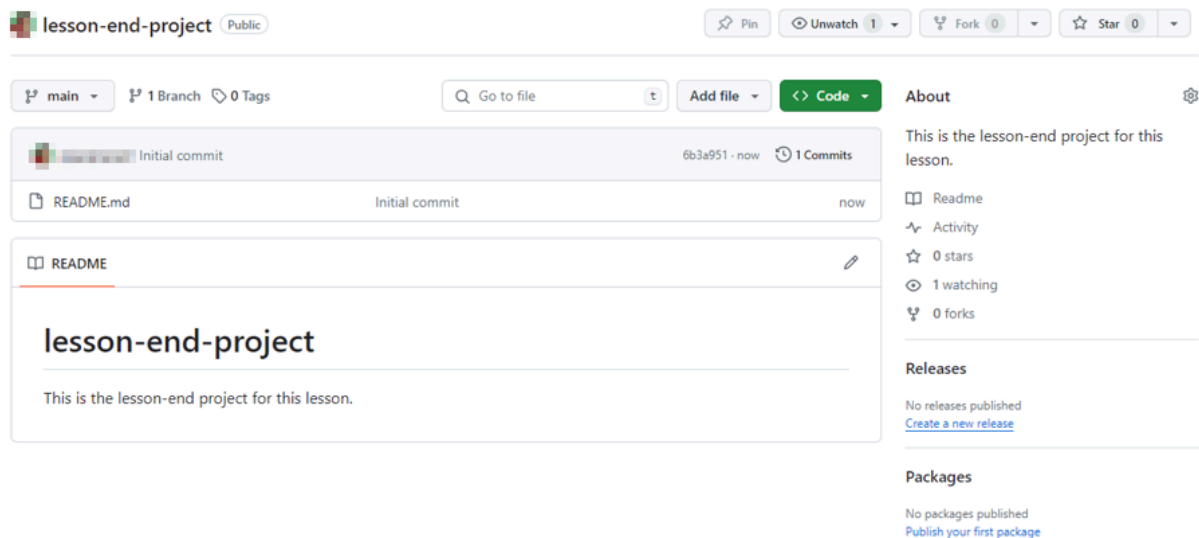
License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

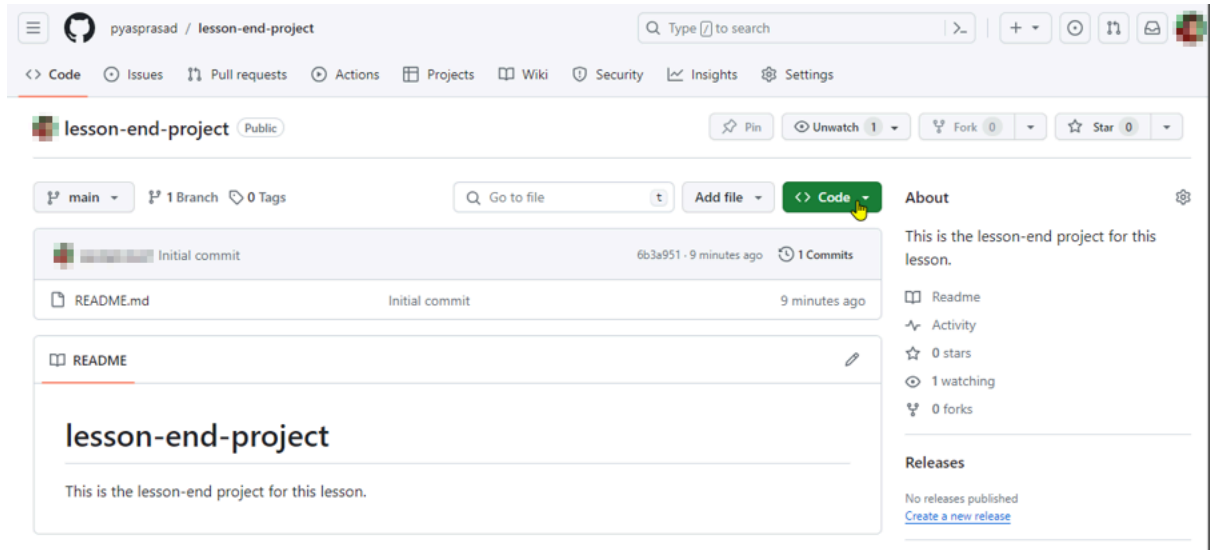


The screenshot shows the GitHub interface for a newly created public repository named "lesson-end-project". At the top, there are buttons for "Pin", "Unwatch" (1), "Fork" (0), and "Star" (0). Below this, the repository name "lesson-end-project" is displayed with a "Public" badge. The main content area shows the "main" branch with 1 branch and 0 tags. A search bar "Go to file" is present. Below the search bar, there are buttons for "Add file" and "Code". The repository's initial commit is shown with a commit hash "6b3a951" and the message "Initial commit". A list of files is shown, including "README.md" and "Initial commit". The "README" file is selected, showing its content: "lesson-end-project" and "This is the lesson-end project for this lesson." On the right side, there is an "About" section with a description: "This is the lesson-end project for this lesson." Below this, there are links for "Readme", "Activity", "0 stars", "1 watching", and "0 forks". There is also a "Releases" section with the text "No releases published" and a link "Create a new release". At the bottom, there is a "Packages" section with the text "No packages published" and a link "Publish your first package".

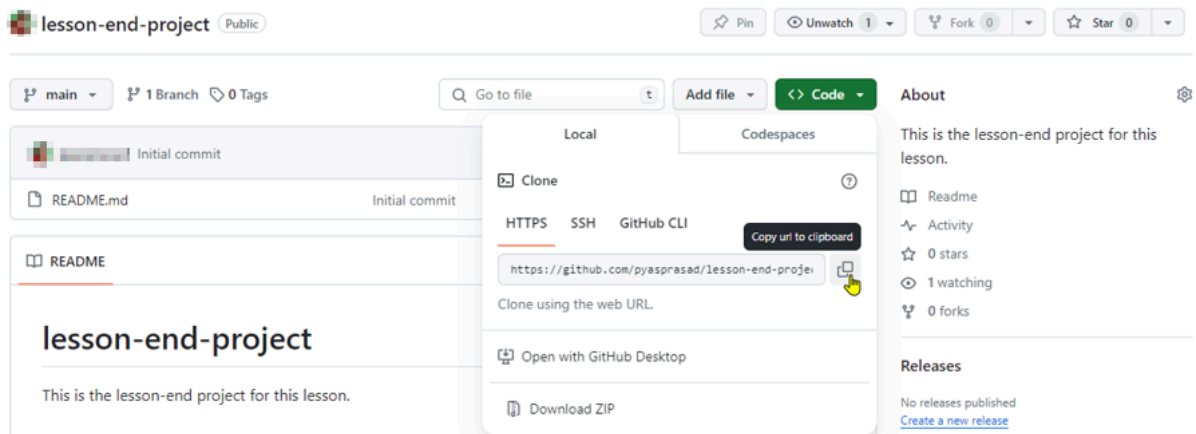
The remote GitHub repository is created.

Step 2: Clone the GitHub repository

2.1 Open the created repository and click on the **Code** button



2.2 Click on the copy icon to copy the **HTTPS URL** as shown below:



2.3 Open the terminal tab on your lab and use the following command to clone the repository:

git clone <URL>

```
priyanshurajsim@ip-172-31-28-201:~/Pyas$ git clone https://github.com/pyasprasad/lesson-end-project.git
Cloning into 'lesson-end-project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
priyanshurajsim@ip-172-31-28-201:~/Pyas$
```

Note: Replace the URL in the command with the copied **HTTPS URL** from the GitHub repository

By following these steps, you have successfully created and cloned a GitHub repository for understanding the fundamentals of repository management and version control using Git and GitHub.