**Placement Empowerment Program**



***Cloud Computing and DevOps Centre***

Deploy Your Static Website Using GitHub Pages Host your local Git repository’s static website directly using GitHub Pages

Name: Kirubalini S I Department: IT

**INTRODUCTION**

GitHub Pages is a free hosting service provided by GitHubthat allows you to deploy static websites directly from arepository. It is an excellent solution for hosting personalprojects, documentation, portfolios, and other simplewebsites. Using GitHub Pages, you can make your localstatic website live on the internet with minimal setup.

**OVERVIEW**

The deployment process involves:

Creating a GitHub repository for your website.

Setting up Git locally on Windows and linking it to GitHub.

Uploading your static website files (HTML, CSS, JavaScript).

Enabling GitHub Pages from the repository settings.

Accessing your live website via a GitHub-provided URL.

**OBJECTIVES**Set up a local Git repository on Windows.

Push a static website to GitHub.

Enable GitHub Pages for hosting.

Access and manage your website online



**IMPORTANCE**

**Free Hosting:** No cost for static site deployment.

**Version Control:** Easy tracking of website changes with Git.

**Fast and Reliable:** Hosted on GitHub's global CDN for quickaccess.

**Custom Domain Support:** Can be linked to a personal domain.

**No Backend Required:** Ideal for portfolios, projectdocumentation, and personal blogs.

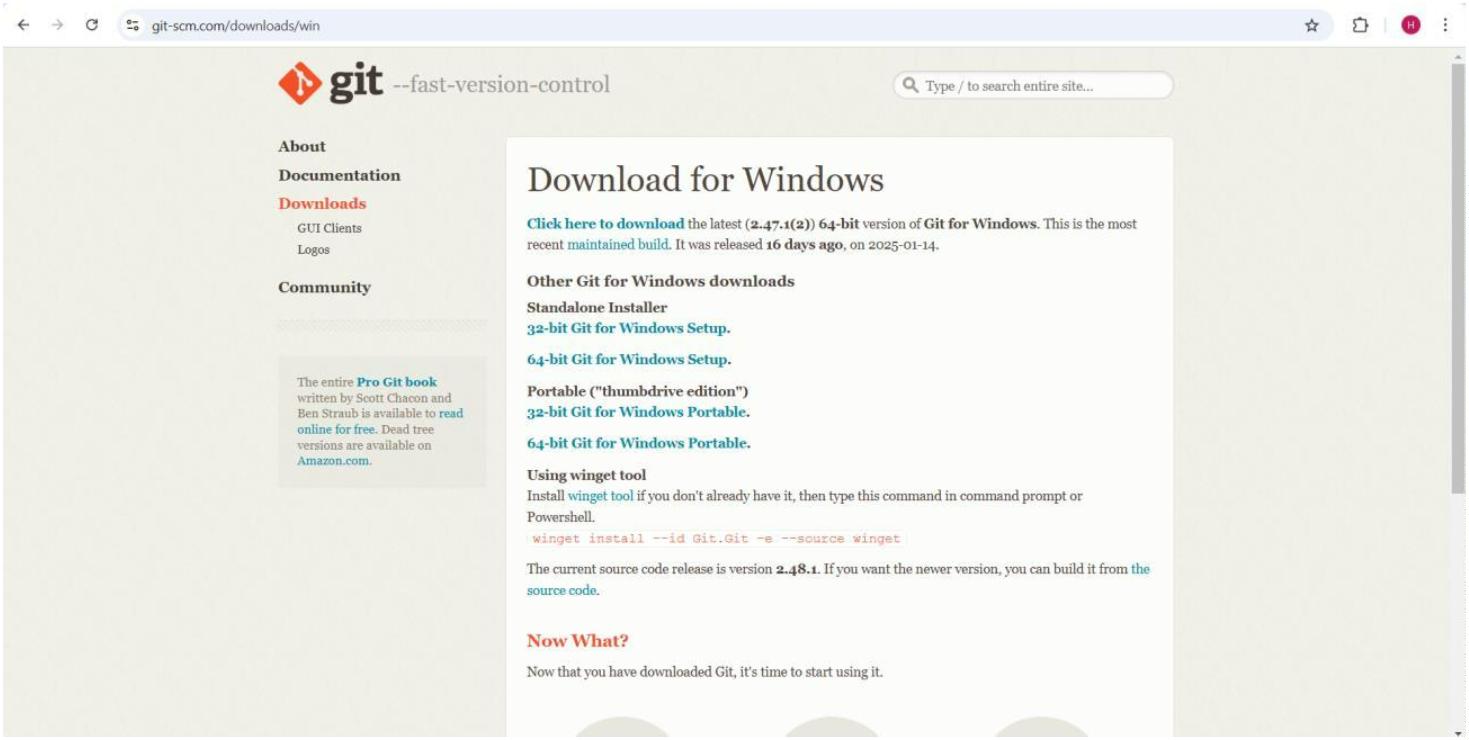
**STEP-BY-STEP OVERVIEW**

**Step 1:**

Install git

1. Download and install Git for Windows from ~~Git for~~

~~Windows~~.



2. During installation, select **Git Bash** as your default

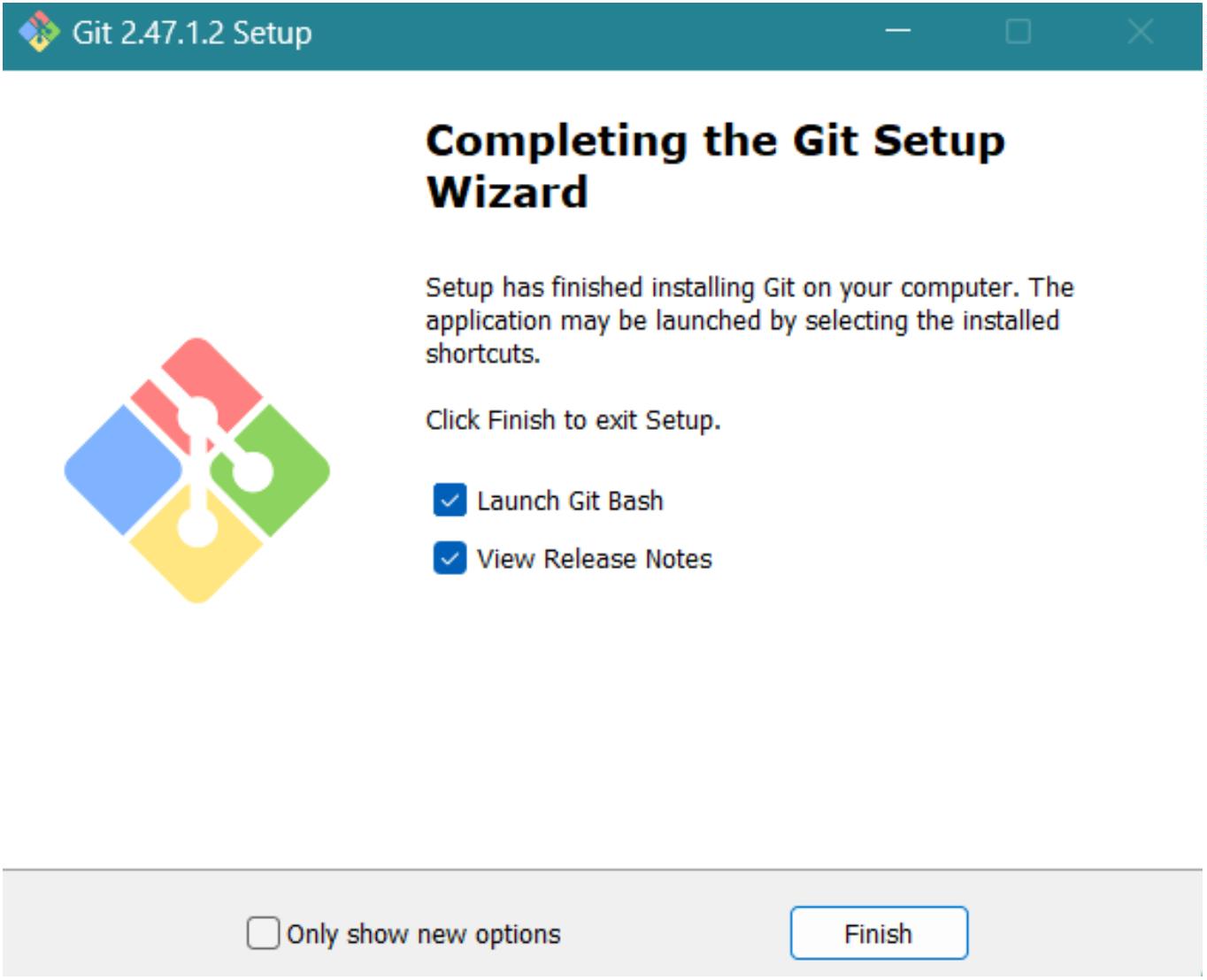
terminal.

**Step 2 :**

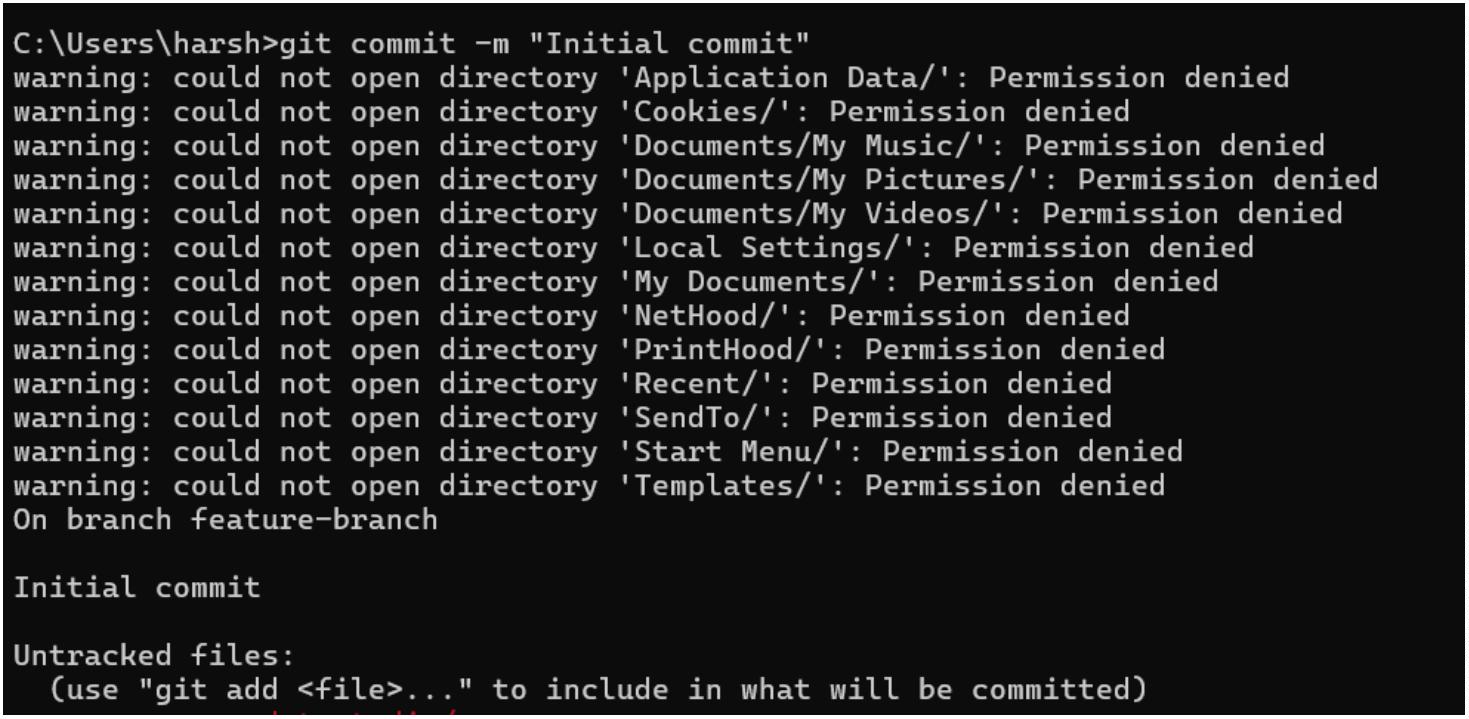
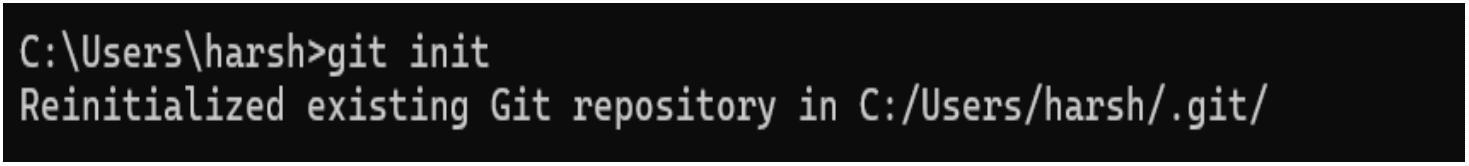
Create a local repository

1. Navigate to your project folder:

cd path\to\your\project



2. Initialize a Git repository:



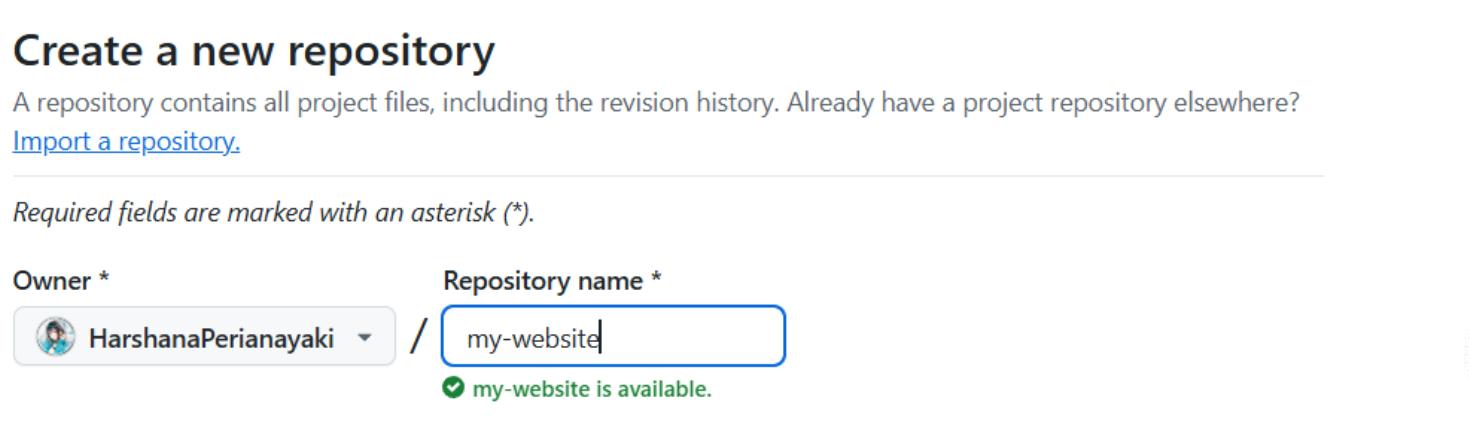
3.Add all files:

3. Commit the changes:

**Step 3 :** Create a Github repository

1. Go to ~~GitHub~~ and sign in. 2. Click on **New Repository**.

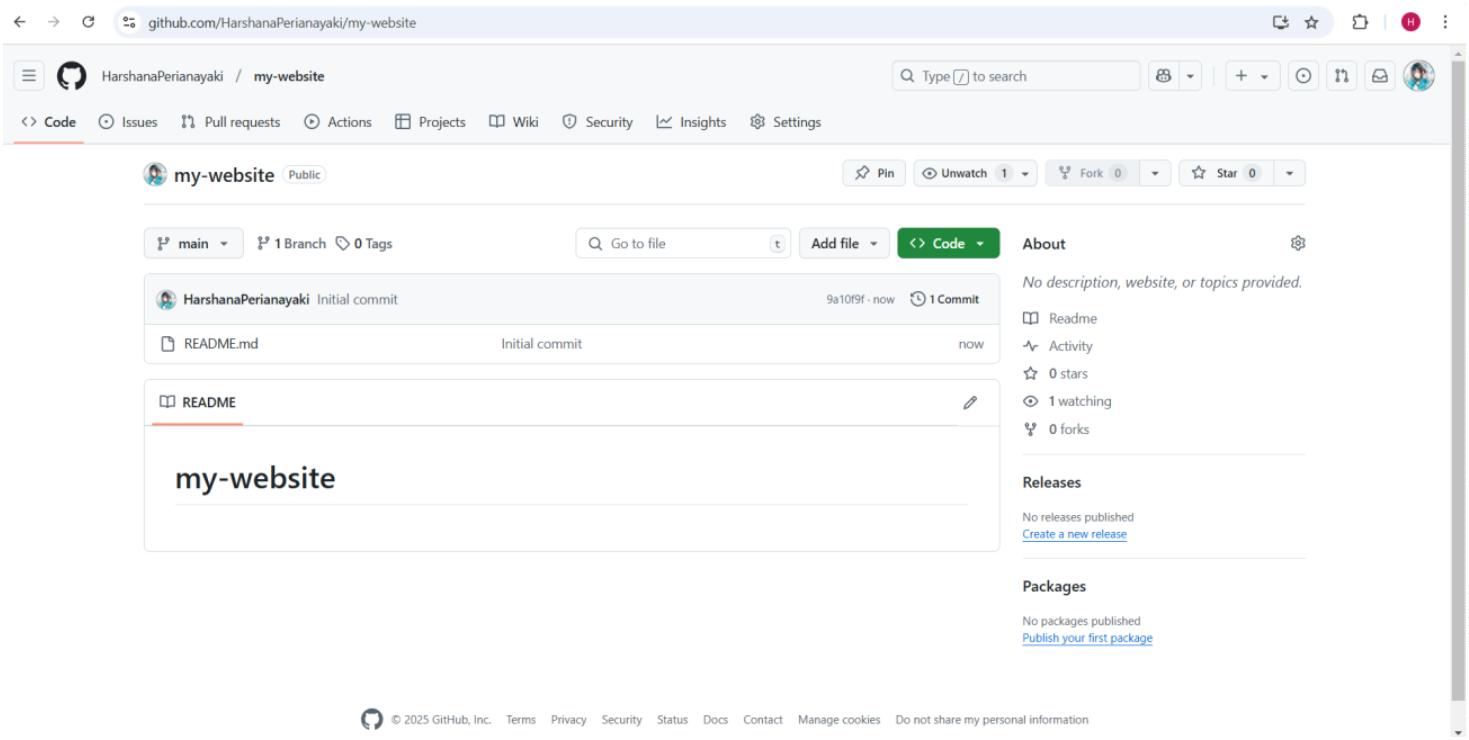
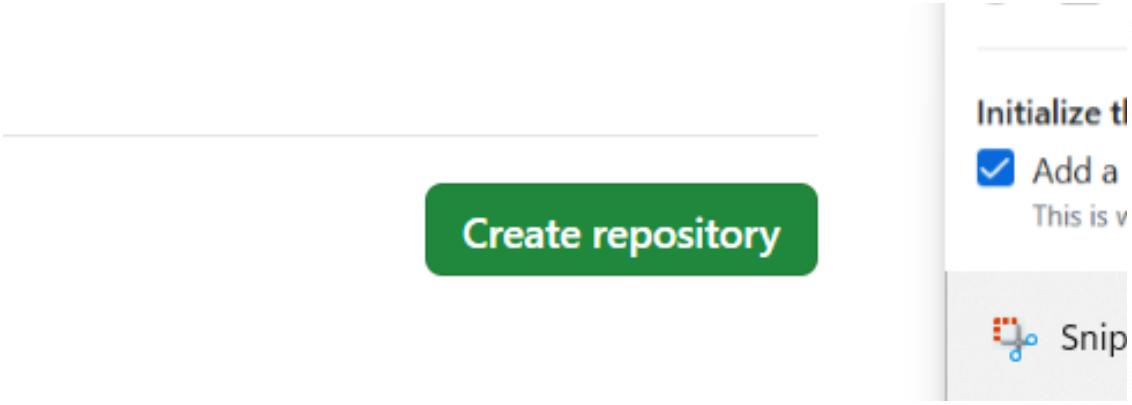
3. Name the repository (e.g., my-website).



4. Select **Public** and **Initialize with a README (optional)**.

5. Click **Create Repository**.

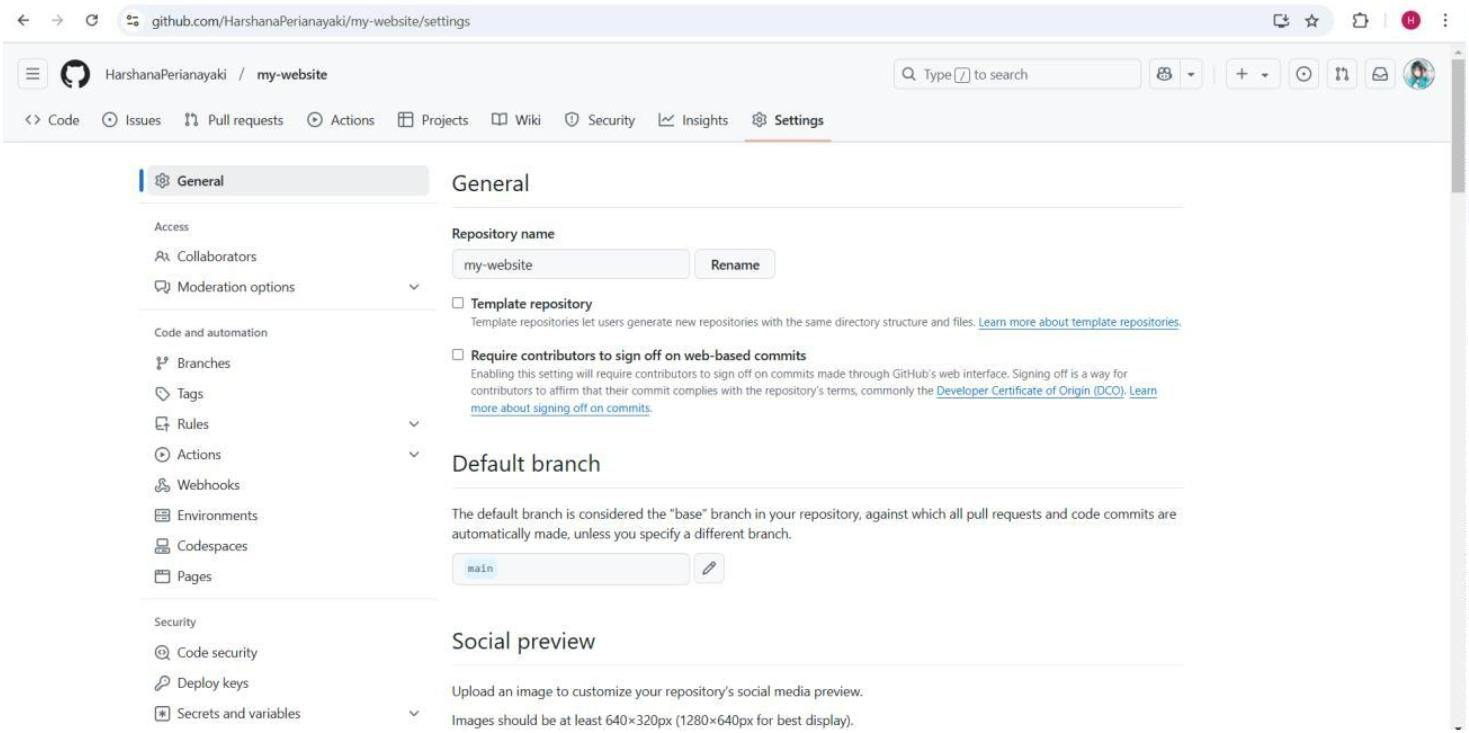
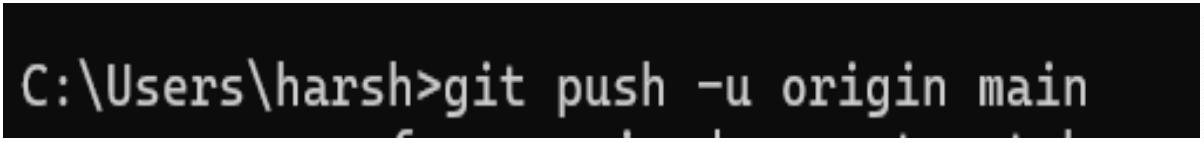
**Step 4 :** Link local repository to GitHub



1. Add the remote GitHub repository:

2.Push your code to GitHub:

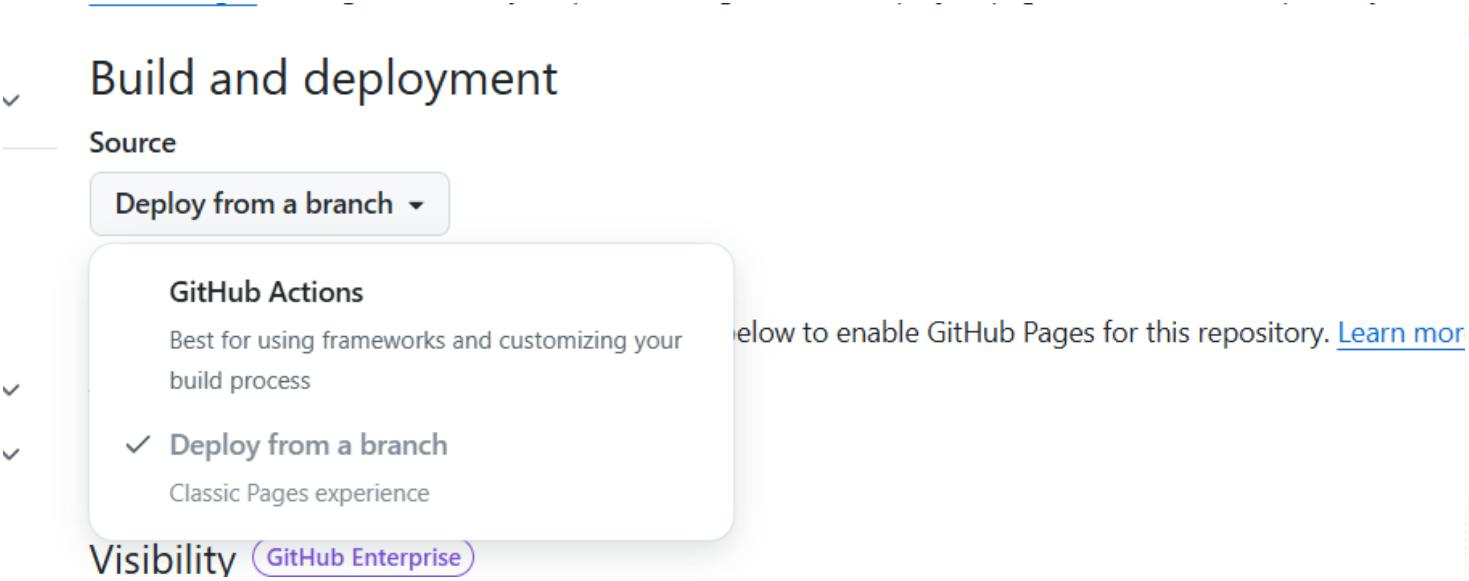
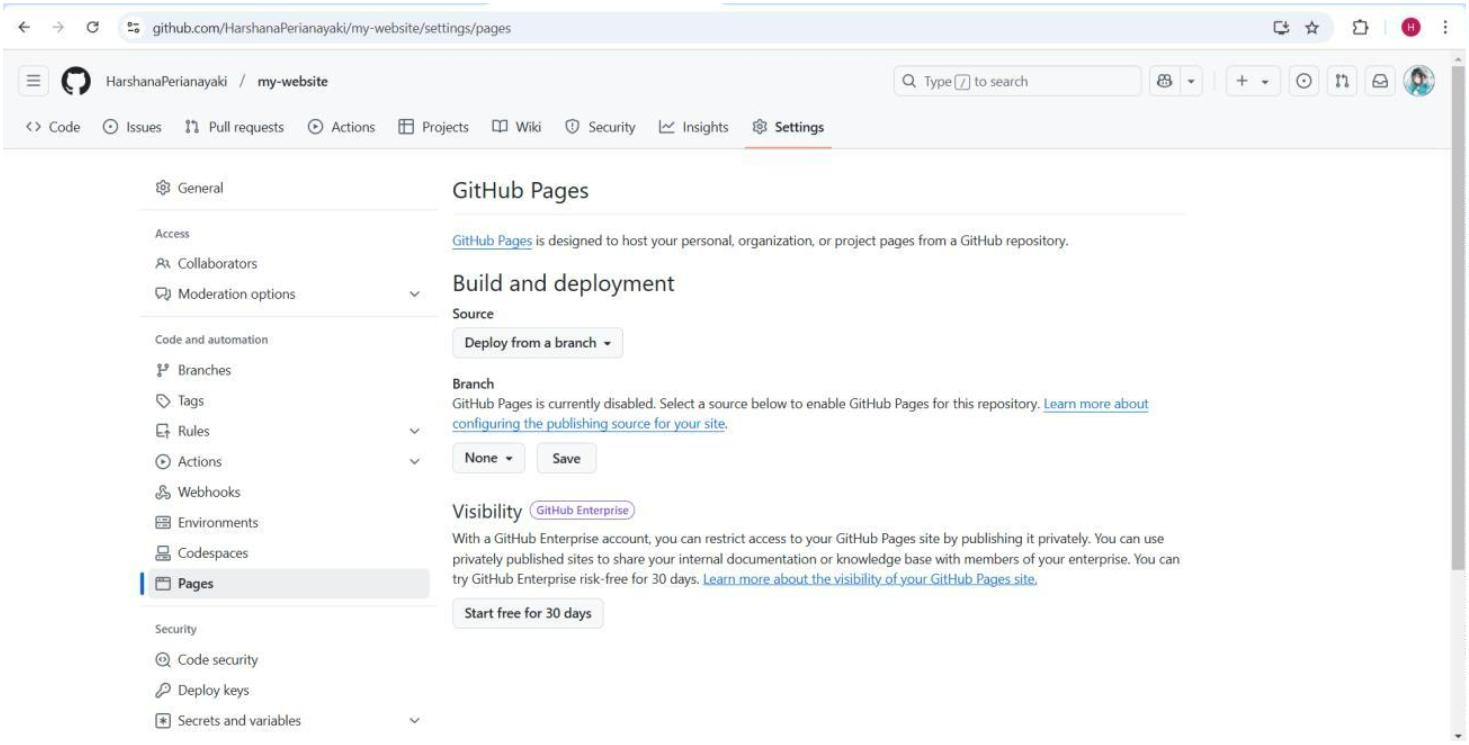
**Step 5 :** Enable GitHub Pages



1. Go to your repository on GitHub.

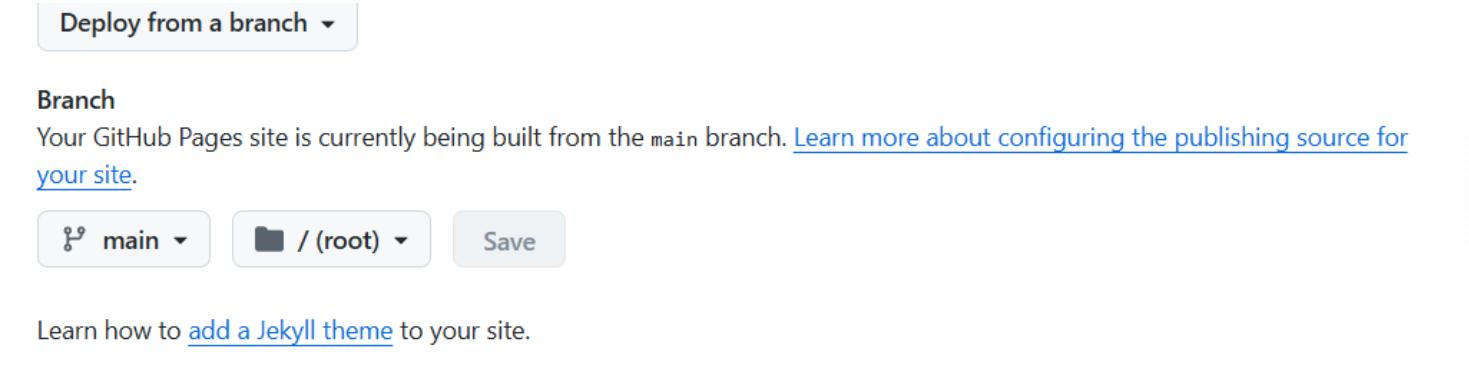
2. Click on **Settings** > **Pages** (left sidebar).

3. Under **Source**, select **Deploy from a branch**.



4. Choose the mainbranch and / (root)directory.

5. Click **Save**.



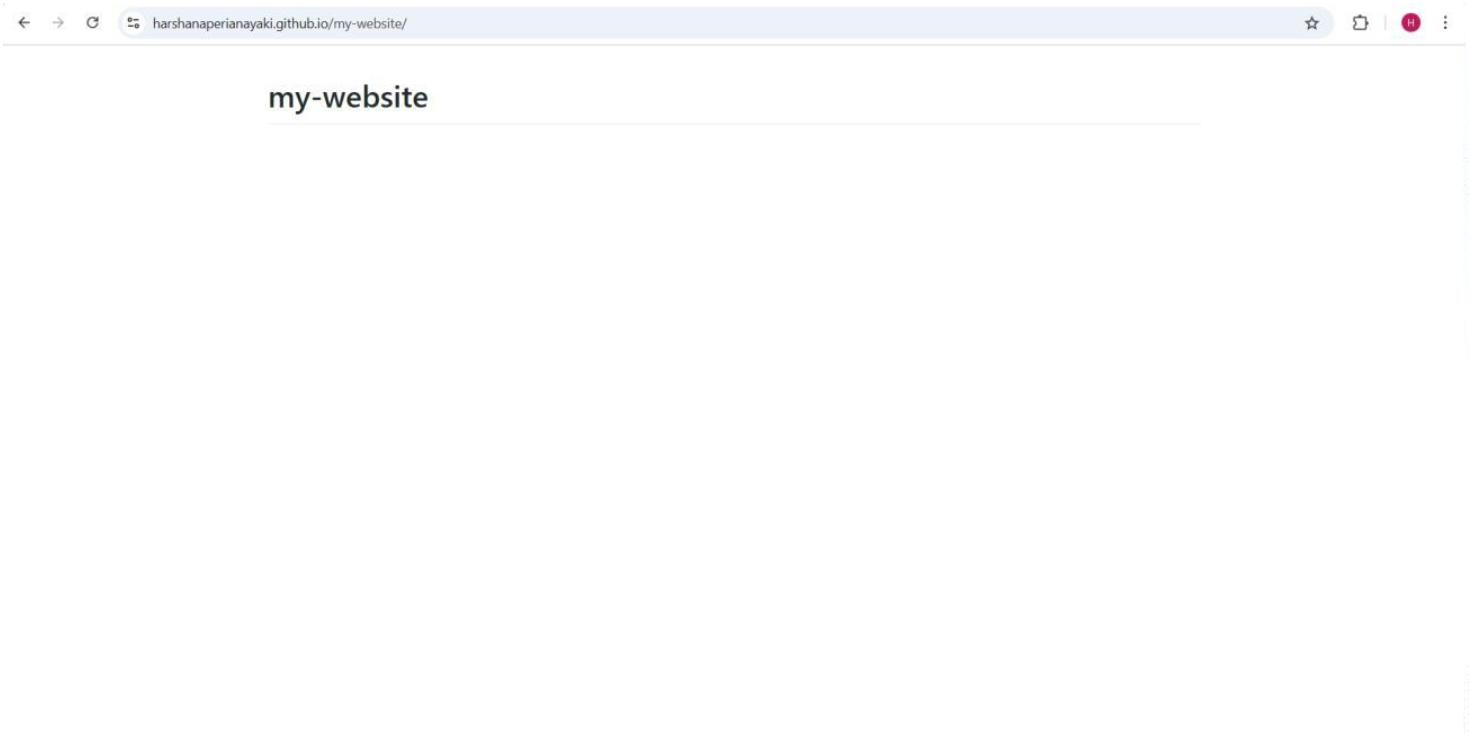
**Step 6 :** Access the Website

After a few minutes, your website will be available at:

**Optional :** Use a Custom Domain

1. Go to **Settings > Pages** and add your custom domain.

2. Update your DNS settings to point to GitHub’s servers.



***OUTCOME***

After completing the deployment, you will:

Have a live static website hosted on GitHub Pages.

Understand Git and GitHub basics for managing websitechanges.

Gain experience in cloud-based deployment for webdevelopment.