**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Set Up a Local Git Repository: Initialize a Git repository locally and version control your static website

Name: AAKASH RAMADURAI B

Department:MECHANICAL ENGINEERING



**Introduction**

Version control is a key part of software development that helps you track changes to your code over time. It lets you manage updates, work with others, and go back to older versions if needed. Git is a popular version control system because it’s fast, flexible, and works well for teams.

In this project, we’ll set up a Git repository to track changes to your static website. This will help you:

- Keep track of file changes.

- Test new features safely.

- Share your project easily.

Using Git is an important step to keep your work organized and reliable, especially when working with others.  
  
**Overview**

Here’s what we’ll do in this setup:

1. **Install Git**: Make sure Git is installed and set up on your computer.

2. **Create a Local Repository**: Start a Git repository in your website’s main folder.

3. **Stage and Commit Files**: Add your files to Git and save a snapshot of your work.

4. **Check the Repository**: Use Git commands to see the status of your files and confirm everything is being tracked.

**Objectives**

By the end of this project, you will:

1. **Learn Version Control Basics**: Understand why Git is important for tracking changes in your projects.

2. **Set Up a Git Repository**: Create a Git repository to manage your static website locally.

3. **Track Changes**: Learn how to save and log changes to your files.

4. **Stay Organized**: Keep your project structured and be able to undo changes if needed.

5. **Get Ready for Teamwork**: Prepare to share your project and work with others using Git.

**Importance of Setting Up a Local Git Repository**

- **Track Changes**: Git keeps a clear history of all changes in your project.

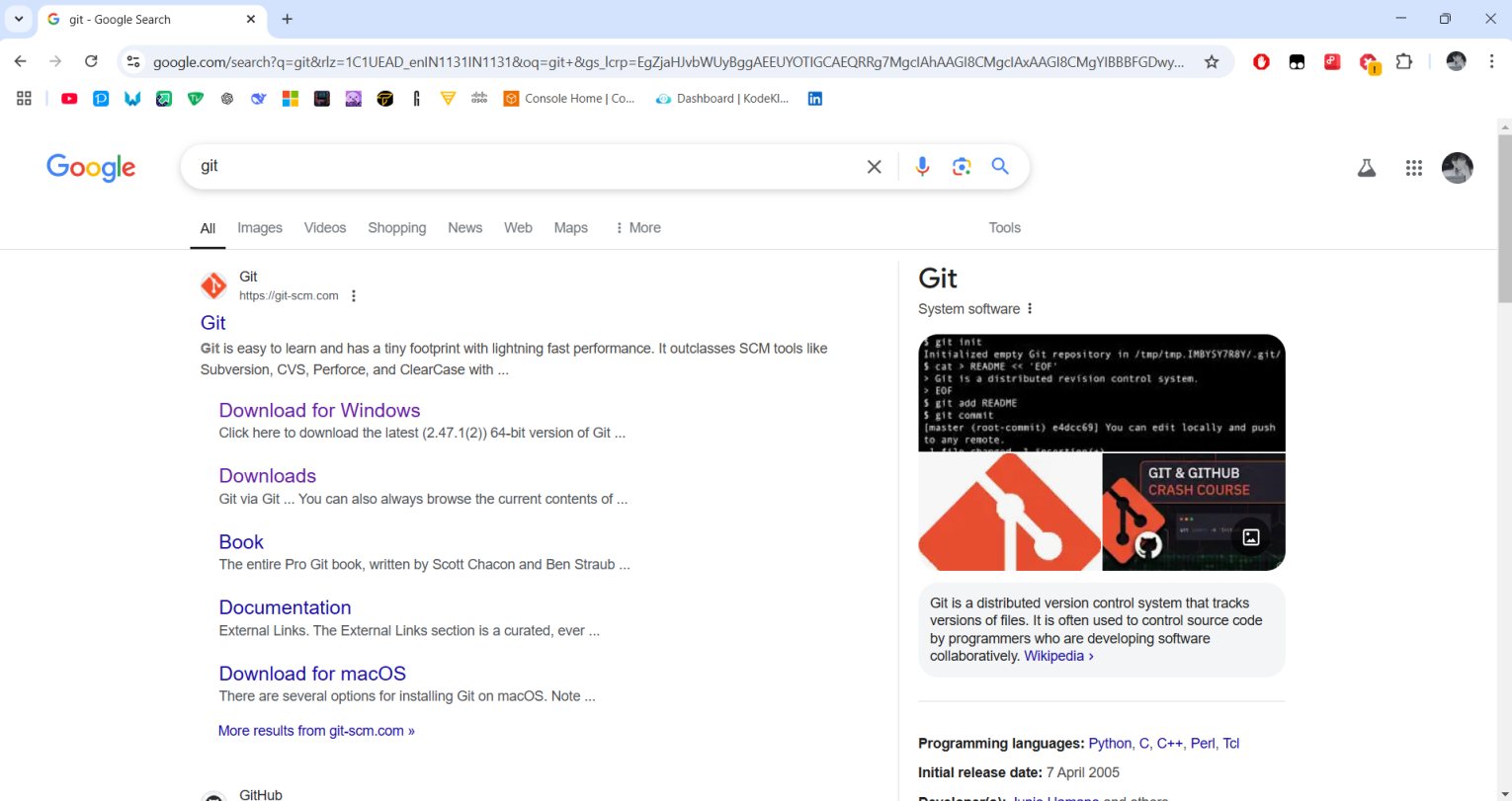
- **Rollback**: Go back to older versions if you make a mistake.

- **Collaboration**: Get ready to work with others by sharing and combining changes easily.

**Step-by-Step Overview**

Step 1:

Search for "Git" in Chrome, download it, and click the "Downloads" option on the website.





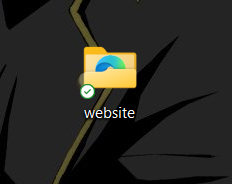
Step 2

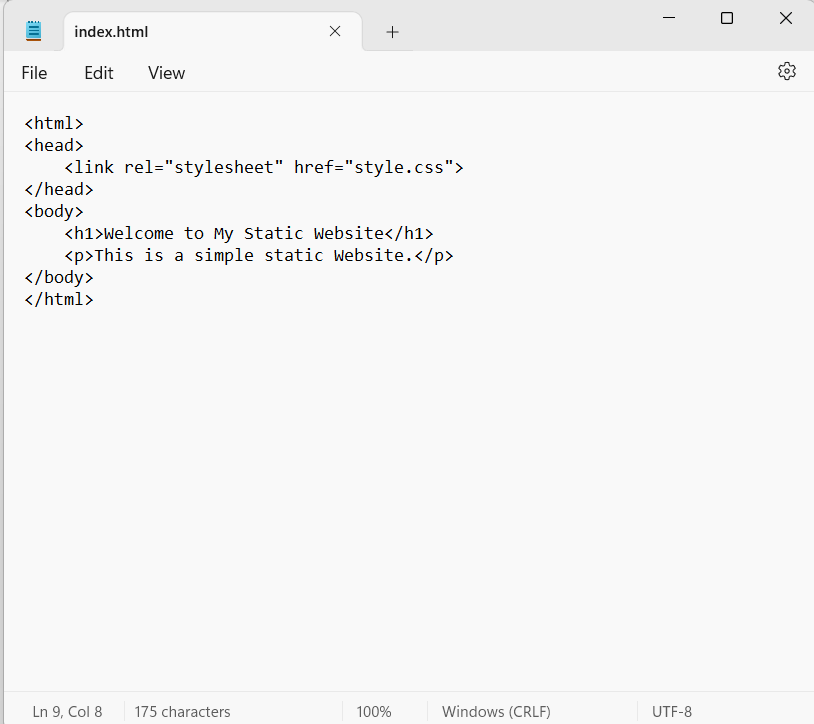
Click the **Windows** option on the download page and follow the installation wizard.

Step 3

In your Desktop Create a folder named website for your static website

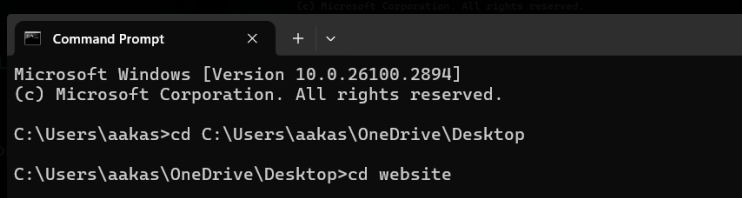
Inside that folder, create a simple HTML file named index.html. You can write some basic HTML





Step 5

Open the Command prompt and set the path to the folder named website we created



Step 6

Now, initialize Git by typing this command:

**git init**

This command will create a .git folder inside your project folder, which tells Git to start tracking your files.



Step 7

Next, we need to tell Git to start tracking your website files.

To tell Git which files to track, use the git add command. If you want to track all the files in your folder, type

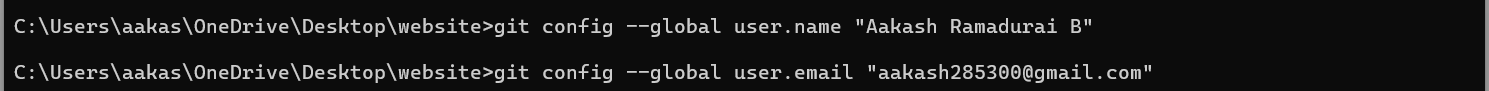
**git add .**

This command adds all the files to Git’s tracking system.



Step 8

Set Up Your Name and Email Globally Git doesn’t know who is making the commit because you haven’t configured your name and email yet. Git uses this information to track who made the changes.



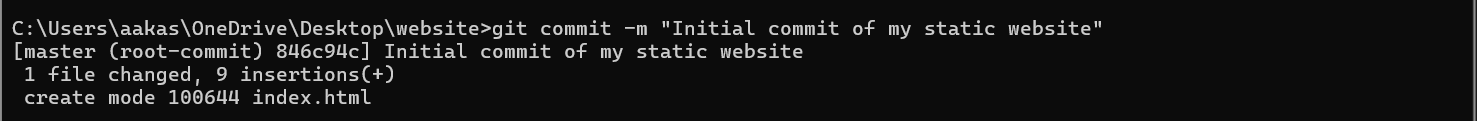
Step 9

Now, we need to save these changes in Git. When you "commit" changes, Git takes a snapshot of your files.

Type the following command to commit your changes:

**git commit -m "Initial commit of my static website"**

The -m flag allows you to add a message about your changes. In this case, we’re saying this is the "initial commit," meaning the first time we’re saving our work.



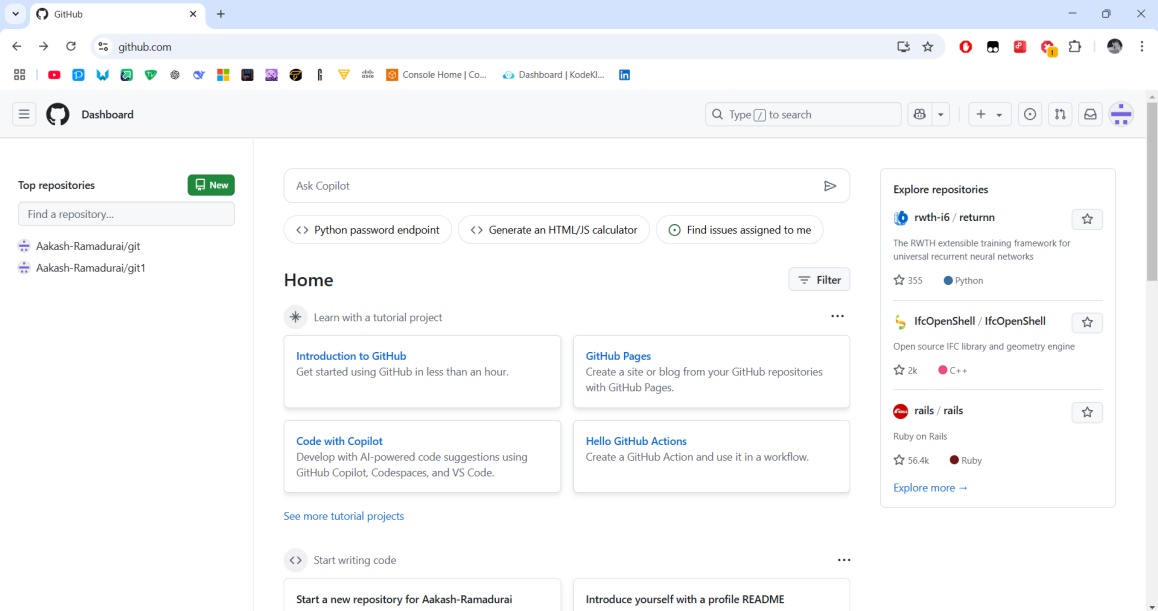
Step 10

**Create a New Repository**:

Once you're logged in, click the green **"New"** button on the top- right of your GitHub homepage to create a new repository.

Give your repository a name, for example, my-website.

Leave the other settings as default, and click **"Create repository"**.



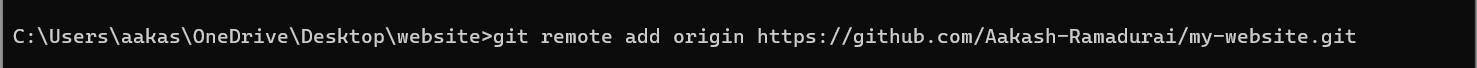
Step 11

**Add the Remote Repository URL to Your Local Repository**:

Go back to your Command Line and type the following:

**git remote add origin https://github.com/yourusername/my-website.git**

Replace yourusername with your GitHub username and my-website with the name of your GitHub repository.



Step 12

The **git branch -M** main command is used to **rename the current branch** to main. Here's what it does:

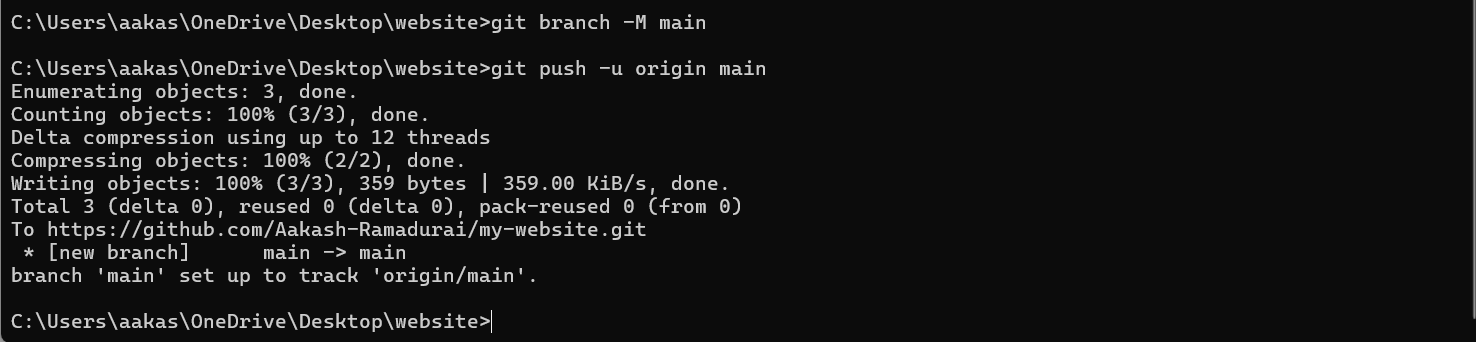
**-M**: This flag forces the renaming, even if a branch named main already exists. It will overwrite the existing main branch.

**main**: This is the new name for the current branch.

image

Step 13

The command git **push -u origin main** is used to **push your local main branch to the remote repository (origin)** and set it as the upstream branch



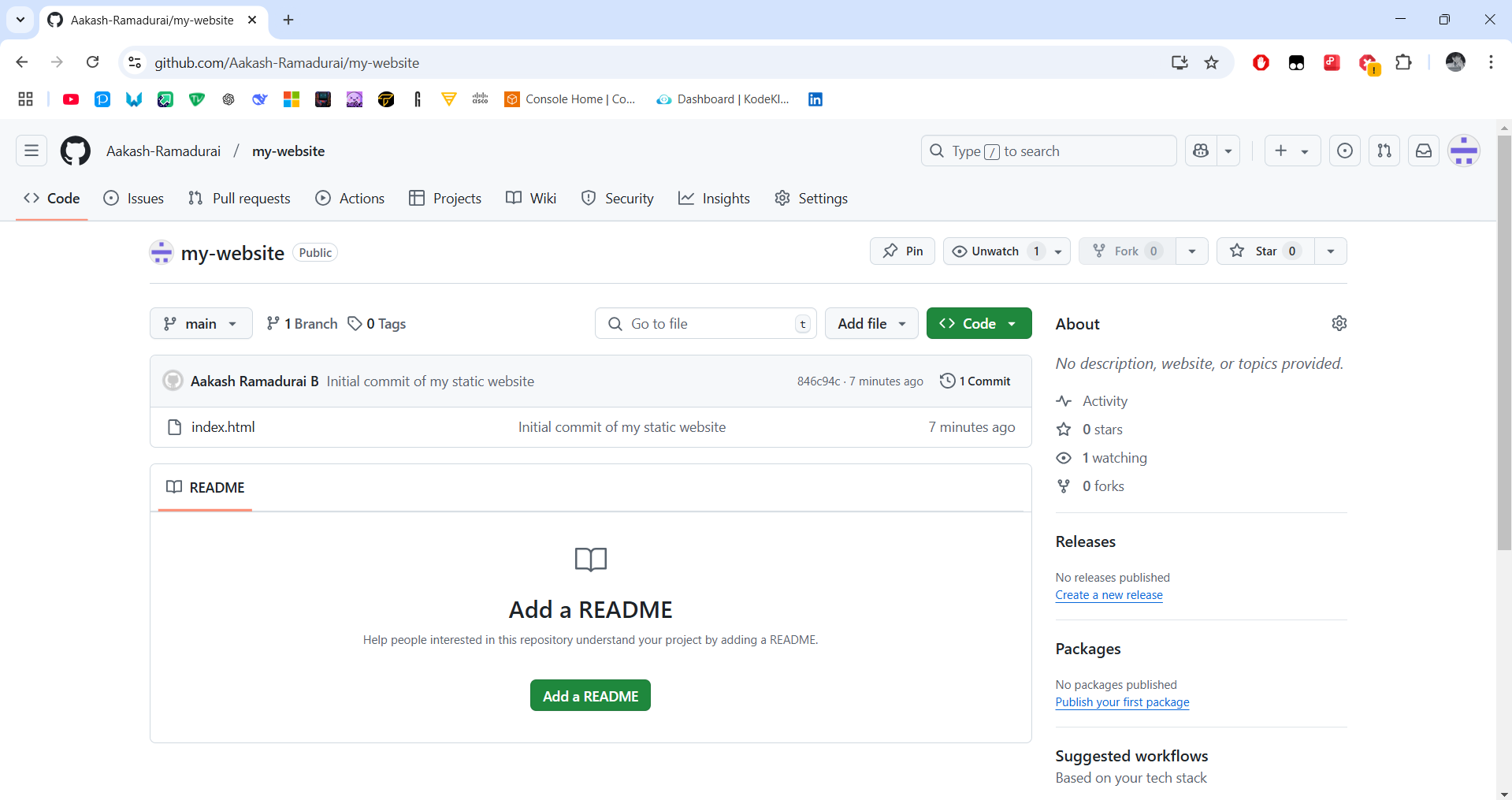
Step 14

Verify Your Files on GitHub

Go to your GitHub Repository:

Open your web browser and navigate to your GitHub repository (e.g., https://github.com/yourusername/my-website).

You should see your website files there!



**Expected Outcome**

By finishing this project, you will:

1. Create a Git repository in your website folder.

2. Use Git to track changes to your website files (like HTML, CSS).

3. Learn basic Git commands: `git init`, `git add`, `git commit`.

4. Save changes locally with clear messages.

5. Get practical experience using Git to manage and track file changes.

THANK YOU!