



## **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: Keshika D Department: ADS



## **Introduction**

Version control is a fundamental practice in software development that allows you to manage changes to your code over time. It provides a systematic way to track updates, collaborate with others, and revert to previous versions if needed. Git is one of the most widely used version control systems, known for its efficiency, flexibility, and distributed na

In this POC, we'll initialize a local Git repository to version control your static website. By doing so, you'll be able to track changes to your project file experiment with new features in a controlled way, and easily share your project with others if needed. Setting up a Git repository is a critical step towards maintaining a structured and reliable workflow, especially for developers a teams working on collaborative projects.

## Overview

Here's what we will cover in this setup:

- 1. Installing Git:** Ensure Git is installed on your system and properly configure
- 2. Creating a Local Repository:** Initialize a Git repository in the root folder of your static website
- 3. Staging and Committing Files:** Add your project files to the staging area a commit them to the repository to save a snapshot of your work.
- 4. Reviewing the evolves. Repository State:** Use Git commands to check the status of your repository and verify that everything is tracked properly.

## Objectives

By the end of this POC, you will:

- 1. Understand the Basics of Version Control:** Gain insight into the importance of Git for managing and tracking changes in your projects.
- 2. Set Up a Git Repository:** Learn how to initialize a Git repository to version control your static website locally.
- 3. Track Changes Effectively:** Understand how to stage and commit file ensure every change is logged.
- 4. Organize Your Project:** Maintain a clean and structured workflow for yo static website, with the ability to roll back changes when needed.
- 5. Prepare for Collaboration:** Lay the groundwork to share your repository and collaborate with others using Git when required

# Importance of Setting Up a Local Git Repository

**Track Changes:** Git records all modifications, ensuring a clear history of your project.

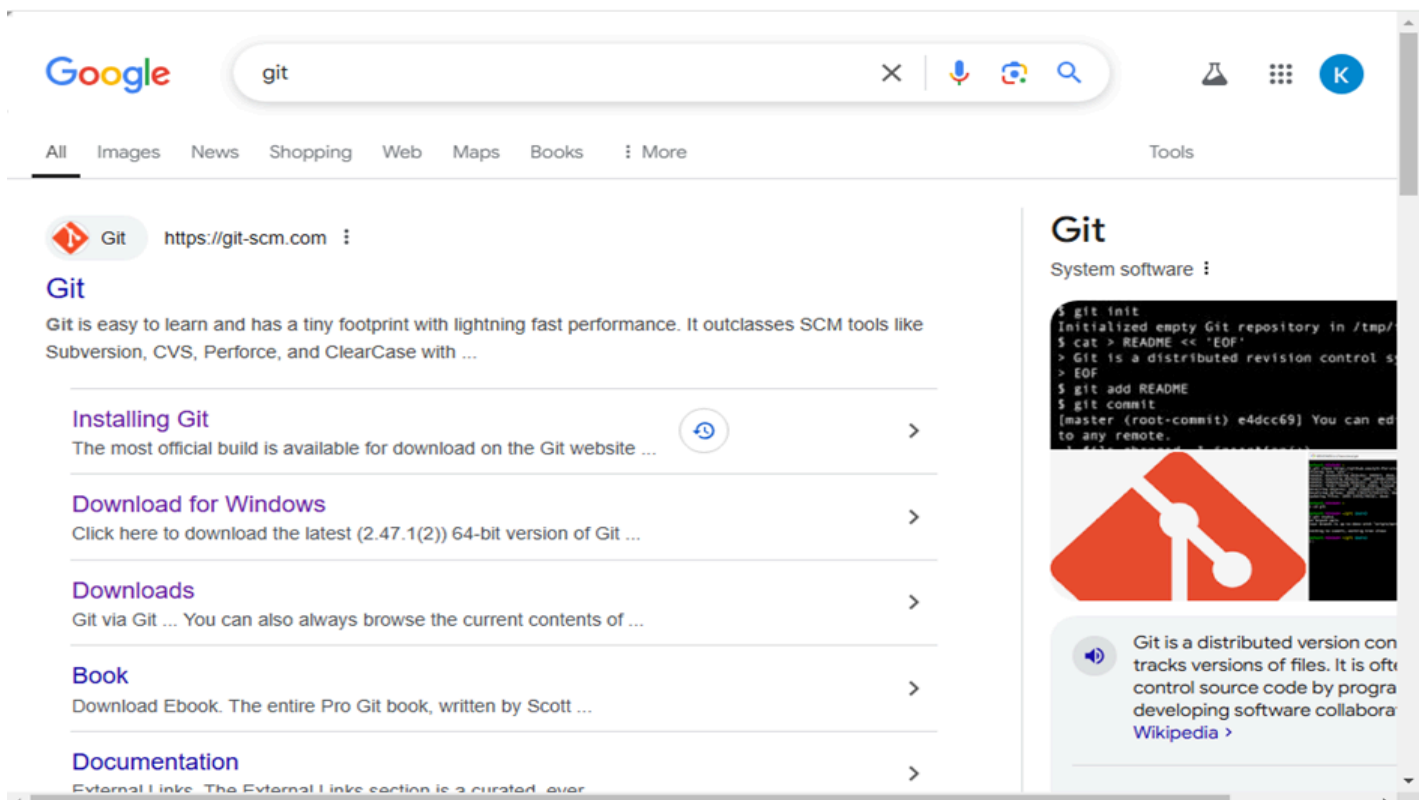
**Rollback:** Easily revert to previous versions to recover from mistakes.

**Collaboration:** Prepares your project for team work, enabling smooth integration of changes.

## 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.

[About](#)[Documentation](#)[Downloads](#)[GUI Clients](#)[Logos](#)[Community](#)

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Download for Windows

[Click here to download](#) the latest (2.47.1(2)) 64-bit version of **Git for Windows**. This is the most recent [maintained build](#). It was released **19 days ago**, on 2025-01-14.

### Other Git for Windows downloads

#### Standalone Installer

[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup.](#)

#### Portable ("thumbdrive edition")

[32-bit Git for Windows Portable.](#)

[64-bit Git for Windows Portable.](#)

#### Using winget tool

Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version **2.48.1**. If you want the newer version, you can build it from [the source code](#).

### Now What?

## 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



```
<html>
<head>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Welcome To my Static Website</h1>
  <p>This is a simple static website.</p>
</body>
</html>|
```

## Step 5

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

```
C:\WINDOWS\system32\cmd.exe
```

```
Microsoft Windows [Version 10.0.19045.5371]  
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\TEMP.DESKTOP-S0M6S0K.005>E:
```

```
E:\>cd E:\Keshika\website
```

```
E:\Keshika\website>
```

## 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 file

```
E:\Keshika\website>git init  
Reinitialized existing Git repository in E:/Keshika/website/.git/  
E:\Keshika\website>
```

## Step 7

Next, we need to tell Git to start tracking your website file

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.

```
E:\Keshika\website>git add .
```

## 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.

```
Reinitialized existing Git repository in E:/Keshika/website/.git/
```

```
E:\Keshika\website> git config --global user.name "Keshika"
```

```
E:\Keshika\website>git config --global user.email "keshikad14@gmail.com"
```

## Step 9

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

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'



saying this is the "initial commit," meaning the first time we're saving our work.

```
E:\Keshika\website>git commit -m "Initial commit of my static website"
On branch master
```

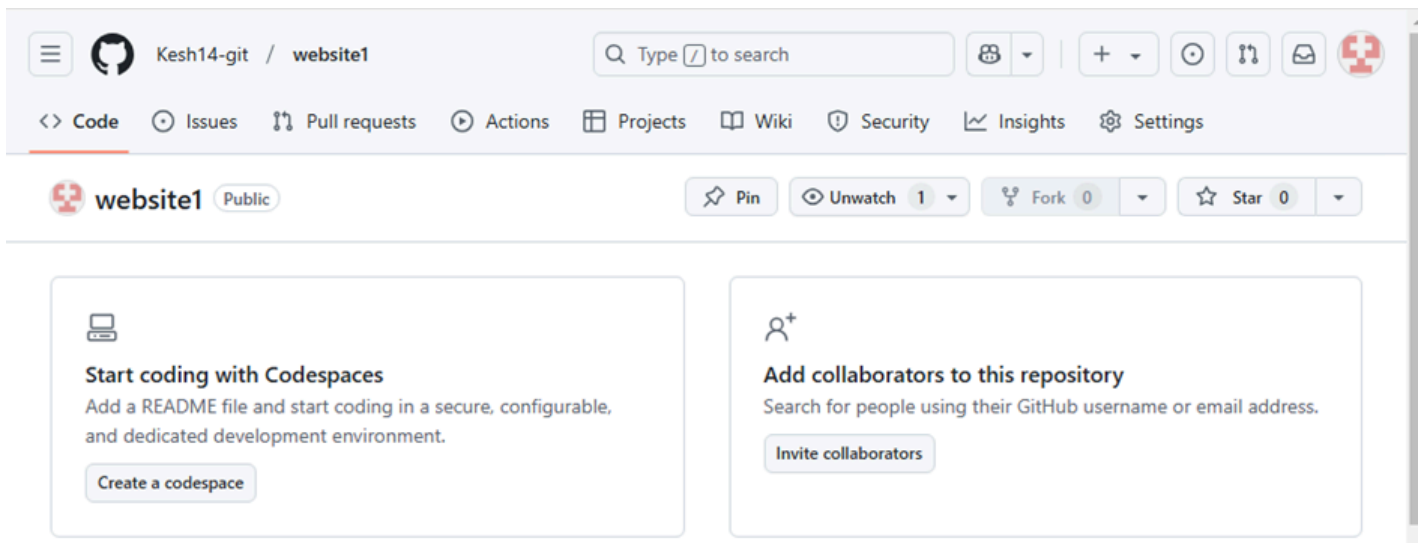
## Step 10

### 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 exist It will overwrite the existing main branch.

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

```
E:\Keshika\website>git remote add origin https://github.com/Kesh14-git/website1
```

```
E:\Keshika\website>git checkout -b main
Switched to a new branch 'main'

E:\Keshika\website>git -u origin main
unknown option: -u
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
          [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
          [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]
          <command> [<args>]

E:\Keshika\website>
```

## 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 refer the above image.

## 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!

```
E:\Keshika\website>file:///E:/Keshika/website/index1.html
'file:' is not recognized as an internal or external command,
operable program or batch file.

E:\Keshika\website>Is
'Is' is not recognized as an internal or external command,
operable program or batch file.

E:\Keshika\website>git add .

E:\Keshika\website>git commit -m "Added website files"
[main (root-commit) 6370fa1] Added website files
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index1.html

E:\Keshika\website>git push -u origin main
info: please complete authentication in your browser...
```

## Expected Outcome

By completing this PoC of setting up a local Git repository, you will: 1.

Successfully initialize a Git repository in your local static website folder.

2. Track changes made to your website files (HTML, CSS, etc.) using G version control.
3. Understand the basic Git commands (git init, git add, git commit) for version control.
4. Commit your changes locally with a descriptive commit message.
5. Gain hands-on experience with Git and how it helps manage and track

bi file  
website file change