

Assignment No. 1

Git Installation & Configuration

Objective

To install Git on a Windows system and configure basic user details such as username and email ID to enable efficient version control and collaboration.

Introduction to Git – Version Control System

Git is a **distributed version control system (DVCS)** designed to track and manage changes in source code during the software development lifecycle. It helps developers maintain a complete history of a project, collaborate with teams, and restore previous versions of files whenever required.

Instead of storing only file differences, Git captures **snapshots** of the entire project at various points in time. Each snapshot represents the complete state of the project, which makes Git fast, reliable, and efficient.

A major advantage of Git is its **distributed architecture**. Every developer has a full copy of the repository on their local machine, including the entire commit history. This allows developers to work offline, make local commits, and later synchronize their work with remote repositories when internet access is available.

Key Features of Git

- Branching and merging for independent feature development
- Commit history to track changes and contributors
- Rollback and revert options to undo mistakes
- Easy collaboration using remote platforms such as GitHub

Due to its speed, flexibility, and robustness, Git is widely used in modern software development for version control and teamwork.

Git Workflow

Git follows a structured workflow that consists of four main stages:

1. Working Directory

- The directory where files are created, edited, or deleted
- Changes are made directly to files
- Git does not track changes automatically at this stage

2. Staging Area

- Acts as a temporary holding area before committing
- Files are added using the git add command
- Allows selective inclusion of files in a commit

3. Local Repository

- Stores committed changes permanently
- Each commit is saved with a unique commit hash
- Maintained on the developer's local system

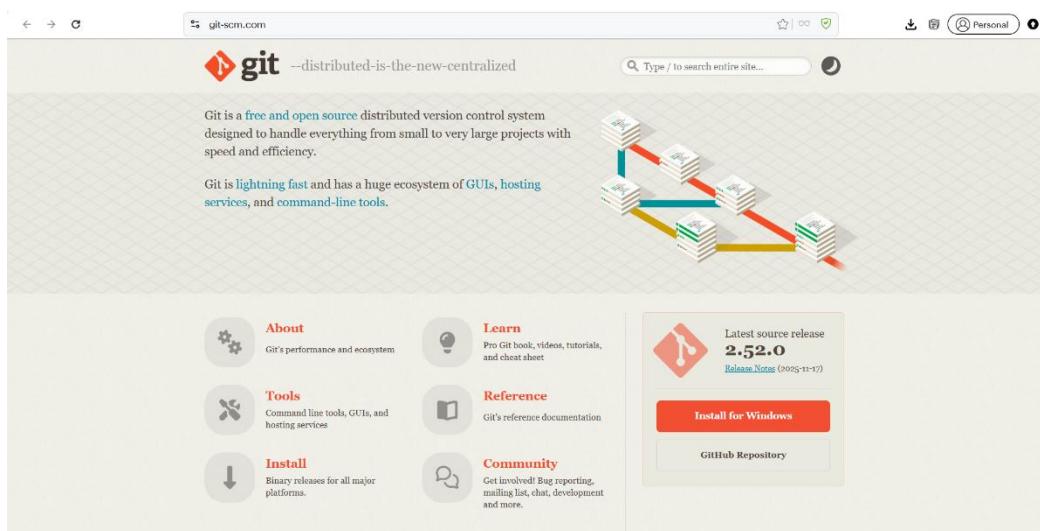
4. Remote Repository

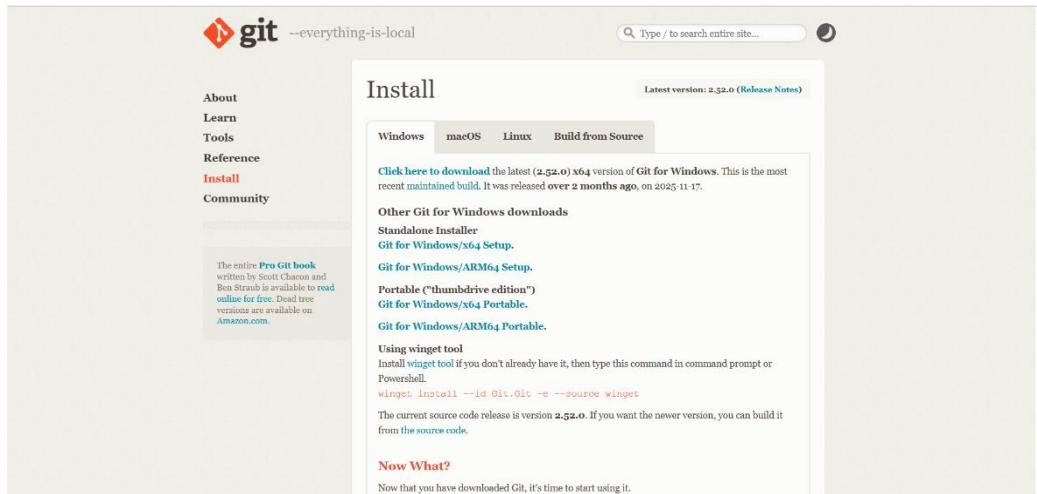
- Hosted on platforms like GitHub
- Used for sharing and collaboration
- Developers push and pull changes to synchronize work

Git Installation (Windows)

Step 1: Download Git

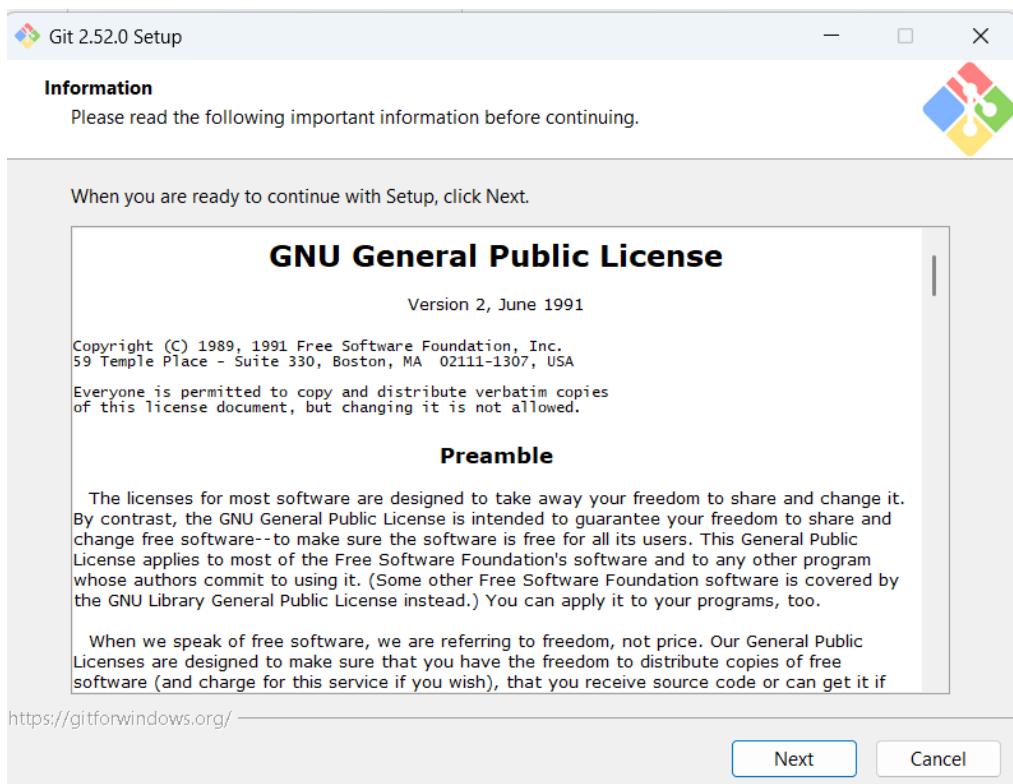
1. Open any web browser
2. Visit  <https://git-scm.com>
3. Click on **Download for Windows**
4. The installer will begin downloading automatically





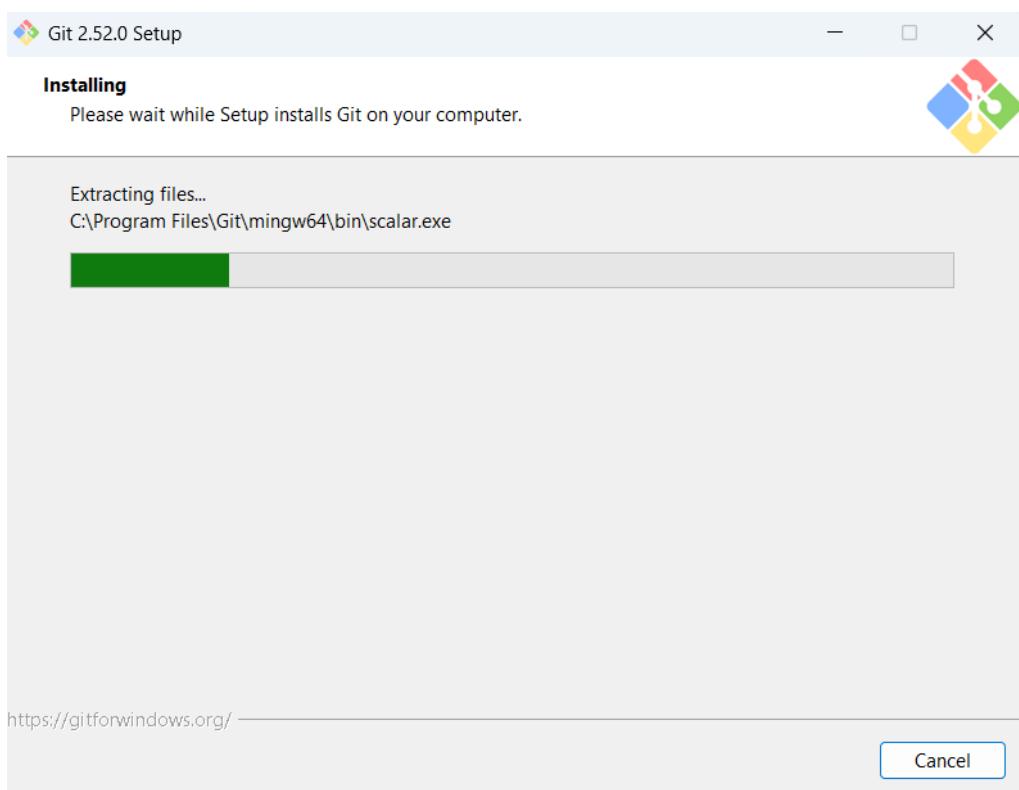
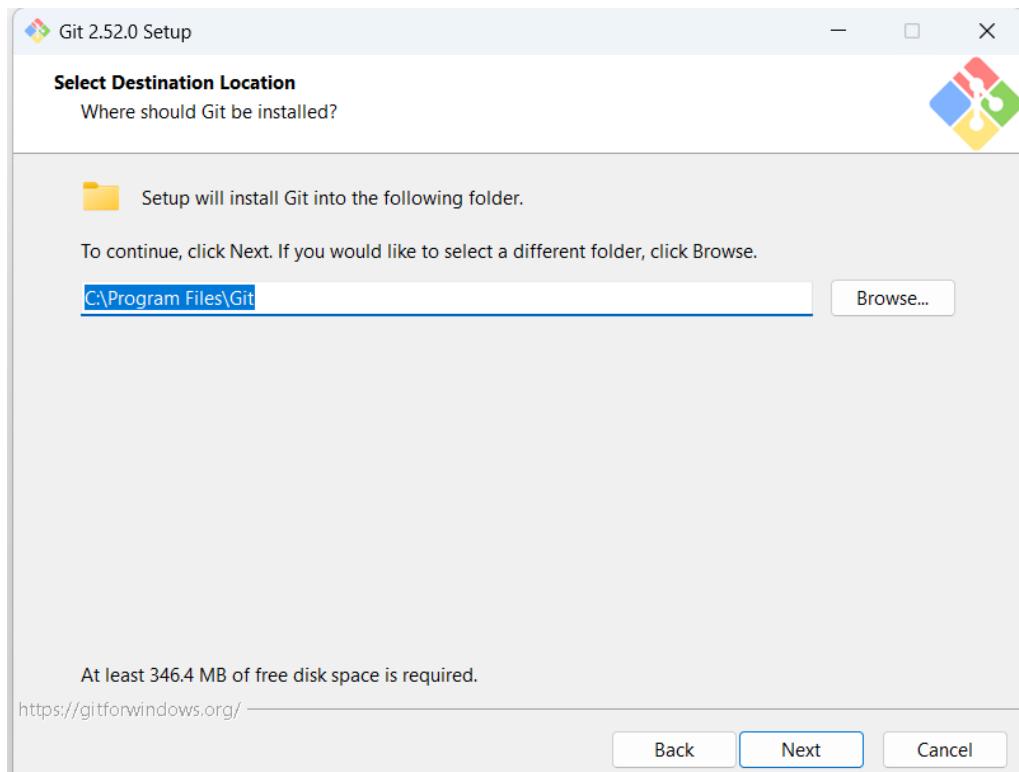
Step 2: Run the Installer

1. Open the downloaded .exe file
2. Click **Next** on the welcome screen



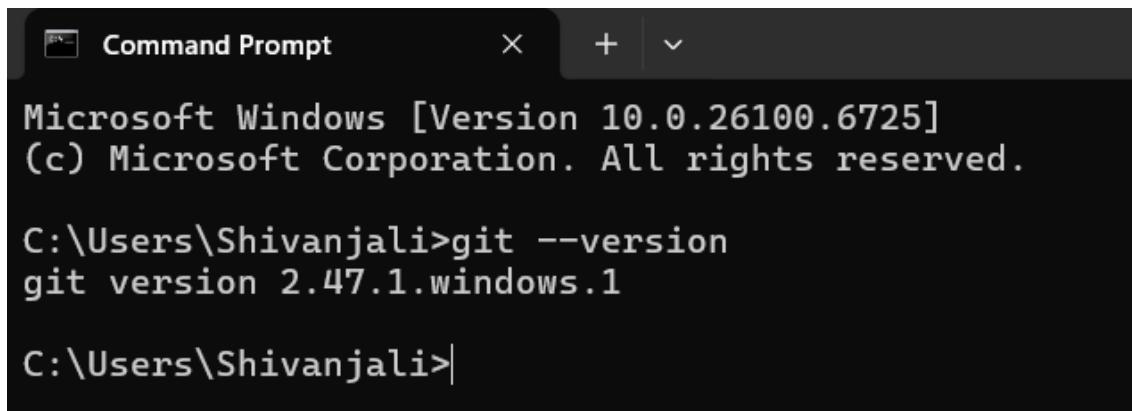
Step 3: Installation Setup

1. Choose the destination folder → Click **Next**
2. Select required components → Click **Next**
3. Continue with default settings unless customization is required



Step 4: Complete Installation

1. Click **Finish** to complete installation
2. Open **Git Bash** to verify installation



```
Command Prompt      X + ▾
Microsoft Windows [Version 10.0.26100.6725]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Shivanjali>git --version
git version 2.47.1.windows.1

C:\Users\Shivanjali>
```

First-Time Git Configuration

After installation, Git must be configured with user details.

Set Username

```
git config --global user.name "Your Name"
```

Set Email ID

```
git config --global user.email your_email@example.com
```

Verify Configuration

```
git config --global --list
```

```
C:\Users\Shivanjali>git config --global user.name "Shivanjali"
C:\Users\Shivanjali>git config --global user.email "bhosaleshivanjali71@gmail.com"
C:\Users\Shivanjali>git config --global --list
core.editor="C:\Users\Shivanjali\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
user.email=bhosaleshivanjali71@gmail.com
user.name=Shivanjali
C:\Users\Shivanjali>
```

1. Difference Between Git and GitHub

| Git | GitHub |
|--------------------------|--------------------------------|
| Version control software | Online hosting platform |
| Installed locally | Cloud-based service |
| Manages code versions | Stores and shares repositories |

Git

GitHub

Mostly command-line based Provides GUI and web interface

Works without internet Requires internet access

Handles local repositories Hosts remote repositories

2. Types of Version Control Systems

Version Control Systems are categorized into three main types:

1. Local Version Control System (LVCS)

- Versions stored on a single local machine
- No collaboration support
- High risk of data loss
- Example: RCS

2. Centralized Version Control System (CVCS)

- Single central server maintains the repository
- Developers commit changes to the server
- Server failure can disrupt work
- Examples: SVN, CVS

3. Distributed Version Control System (DVCS)

- Every user has a full copy of the repository
 - Commits are first made locally
 - Supports better collaboration and backup
 - Example: Git
-

3. What is .gitignore?

.gitignore is a special file used to tell Git which files or directories should be excluded from version tracking.

Purpose of .gitignore

- Avoids committing unnecessary or sensitive files
- Keeps repositories clean and organized
- Prevents large or auto-generated files from being tracked

Commonly Ignored Files

- Log files
- Temporary files
- Dependency folders
- Environment configuration files

Example .gitignore File

```
node_modules/
```

```
.env
```

```
*.log
```

Conclusion

This assignment provided a clear understanding of Git and its role in version control. We learned how to install Git on a Windows system, configure user credentials, and verify the setup. The Git workflow—from working directory to remote repository—was explained in detail.

The assignment also highlighted the differences between Git and GitHub, explained various types of version control systems, and emphasized the importance of the .gitignore file. Overall, this study offered both theoretical and practical knowledge essential for effective code management and collaboration in real-world software development projects.