VASANTDADA PATIL PRATISHTHAN'S COLLEGE OF ENGINEERING AND VISUAL ARTS

Department of Information Technology

Experiment No.2

Aim: To Understand Version Control System/Source Code Management, install git and create a GitHub Account.

The Fundamentals of a Version Control System (VCS)

The systems that teams use to track changes and different code versions are called Version Control Systems (VCS). Just like with anything else, each VCS has its own unique features and comes with its own set of advantages and disadvantages. There are some basics, though, that make a VCS what it is.
First and foremost, they should retain a long-term history of changes made on a project including creation, deletion, edits and more. This should include the author, date and any notes from the changes as well.

Beyond that, they should have a solution for branching and merging new code changes to the main project to allow concurrent work from multiple members of a team. That is the point, after all. The main codebase for a project is called the Trunk or, simply, Main. Branches are created as independent streams of work that can be merged to the Main. All of this supports traceability in projects. In the event that something goes wrong, it’s easier to look back at the changes made over time and connect them to bugs and errors when they appear. Which brings us to the next question, why is version controlso important?

3 Reasons Why VCS is Critical for DevOps

1. **Avoiding dependency issues in modern containerized applications**
Micro services have essentially become the default for the development of new applications, and more and more teams are containerizing monolithic applications as well. With this trend, dependency issues have entered center stage as something that can make or break a project.

For DevOps, finding the balance between moving quickly and maintaining application reliability is crucial. Part of that means cultivating traceability, gaining visibility into the changes made to the code and understanding how those changes affected application performance. Access to the different code iterations can reveal where new changes exposed dependencies that clash with other parts of the code.

2. **Version control is tied to higher DevOps performance**
The annual State of DevOps study found, in 2014, that “version control was consistently one of the highest predictors of performance.” Top performing engineering teams were able to achieve higher throughput – 8x deployment frequency and 8000x faster deployment lead times.

What’s causing the high correlation between version control and high performance? Well, it’s partially related to the ability to more easily view and understand how changes to one part of the code caused problem across the application. But, it has more to do with how the VCS enables coding practices like Continuous Integration and, as a result, Continuous Delivery/Deployment.

3. **Supports building more reliable applications**

It’s not hard to draw the connections between high performing DevOps teams and reliable applications. DevOps’ role in any organization is to enable successful advancements, and because

Although it may seem counterintuitive at first, the first step is usually making smaller changes more frequently (hence the popularity of CI). Just as important, and even more so in an accelerated workflow, is tracking those changes so that everyone is looking at the same thing and so that troubleshooting is easier in the case of failures.

Systems Built for Version Control

The first version (pun intended) of a version control system was the **Concurrent Versions System (CVS)**. It was created to support cooperation between a small team of developers with conflicting schedules. After proving useful on the project, the creator released the code to the public in 1986.

Source Code Management Using Version Control System:

Software is the most vigorous product of Information Technology. The face of software development has changed unprecedentedly and become more custom over a period of time. If we go back 15 years, one can see software were developed for supporting workstations in organizations with centralized database over multiple locations, but rise in international work standard in turn has made software development more competitive and challenging. As a result, software development happens to take place in a collaborative platform – programmers collaborate their code to a central point from multiple locations. How can programmers collaborate? Well, Source code management tools (Version Control System) tackles every single barrier associated with managing source code such as: Integrity, robustness, synchronization, linearity, and revision control. Version control adhere unique functionalities: commit, push/pull to and from code base, snapshots that makes managing source code fairly simple. However conventional version control system can't manage models and there are problems in syncing main repository and local working copy. This work draws shortcoming of 'Git' as a version control system in managing source code and gives insight into Git's association with source code taking line based approach that makes difficult to version control model diagrams with software development. The limitations are identified and the model is proposed to facilitate enhancement in future.

How To Install Git Bash On Windows:

Download Git Bash

- Step 1: Visit the Official Git Bash Website
- Bash Download

Install Git Bash

- Step 3: Run the Installer
- Step 4: Select Destination Location
- Step 5: Select Components
- Step 6: Select Start Menu Folder
- Step 7: Choose the Default Editor used by Git
- Step 8: Adjust your PATH Environment
- Step 9: Choose HTTPS Transport Backend
- Step 10: Configure the Line Ending Conversions
- Step 11: Configure the Terminal Emulator to use with Git Bash
- Step 12: Configuring Extra Options
- Configuring Extra
- Step 14: Wait for Installation
- Step 15: Complete the Git Setup Wizard

Launching Git Bash

Install Git Bash on Windows

In this tutorial, we are going to learn how to install Git Bash on Windows.

Git Bash for Windows is a package that includes git and bash.

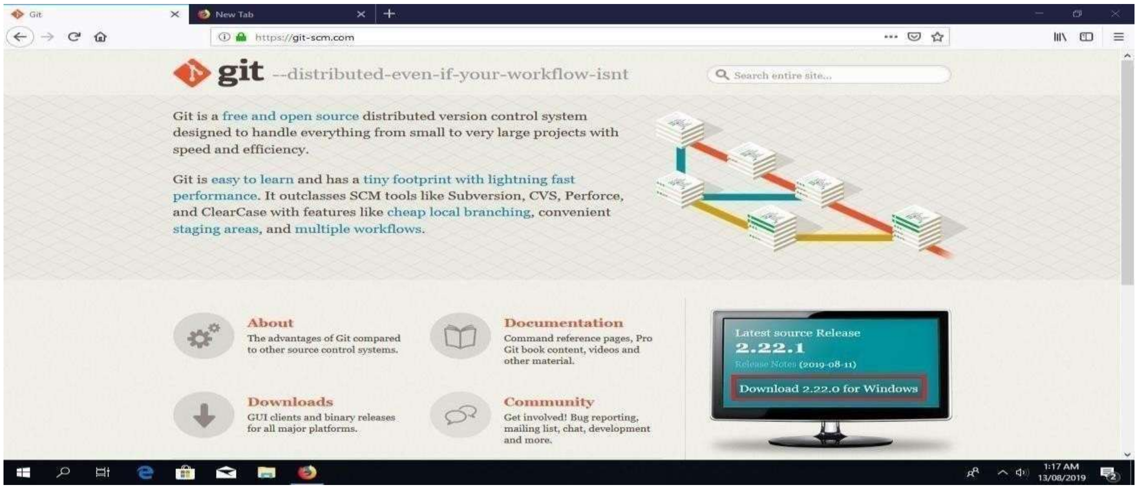
Git is an open-source version control system for tracking source code changes when developing software. It keeps a commit history which allows you to revert to a stable state in case you mess up your code. Gitalso allows multiple developers to collaborate on the same code base.

Bash is a Unix command-line shell. The name is an acronym for the 'Bourne-Again Shell'. It comes withuseful Unix commands like cat, ssh, SCP, etc., which are not usually found on Windows.

Download Git Bash

Step 1: Visit the Official Git Bash Website

Download the latest version of Git Bash from their official website: https://git-scm.com/



--distributed-even-if-your-workflow-isnt

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

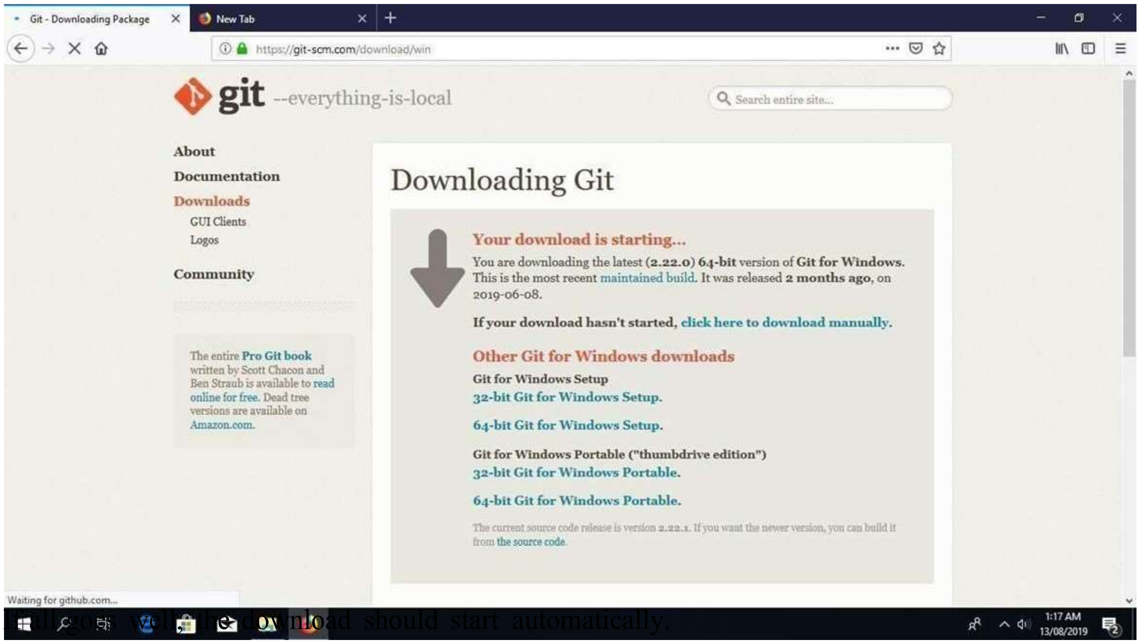
Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.22.1
Release Notes (2019-08-11)
Download 2.22.0 for Windows

Click the "Download for windows" button.

Step 2: Start Git Bash Download Next, you will be redirected to a page that lets you know that you are about to start downloading.



Git - Downloading Package
New Tab
https://git-scm.com/download/win
git --everything-is-local
Search entire site...
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.

Downloading Git

Your download is starting...

You are downloading the latest (2.22.0) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 2 months ago, on 2019-06-08.

If your download hasn't started, click here to download manually.

Other Git for Windows downloads

Git for Windows Setup
32-bit Git for Windows Setup.
64-bit Git for Windows Setup.

Git for Windows Portable ("thumbdrive edition")
32-bit Git for Windows Portable.
64-bit Git for Windows Portable.

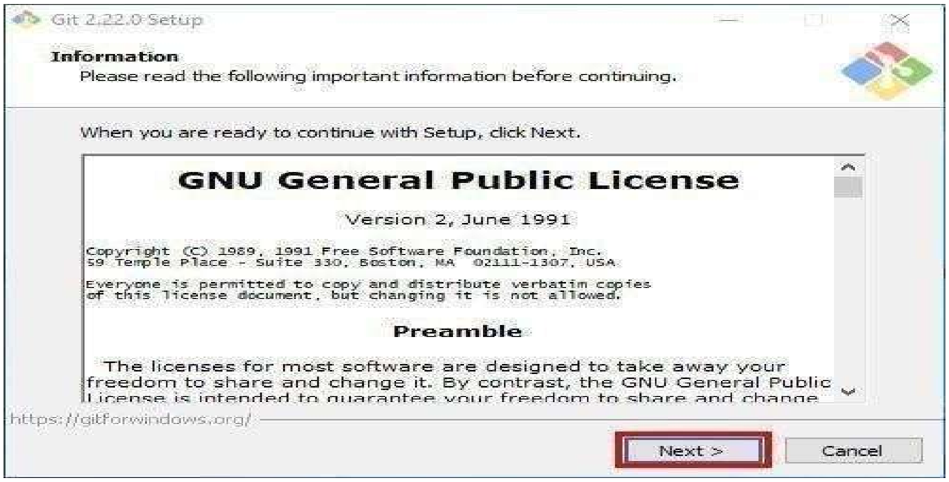
The current source code release is version 2.22.1. If you want the newer version, you can build it from the source code.

If all goes well, the download should start automatically.

Click on “Save File” to start downloading the executable.

Install Git Bash
Step 3: Run the Installer

Once you have downloaded the Git Bash executable, click it to run the installer.



Information
Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

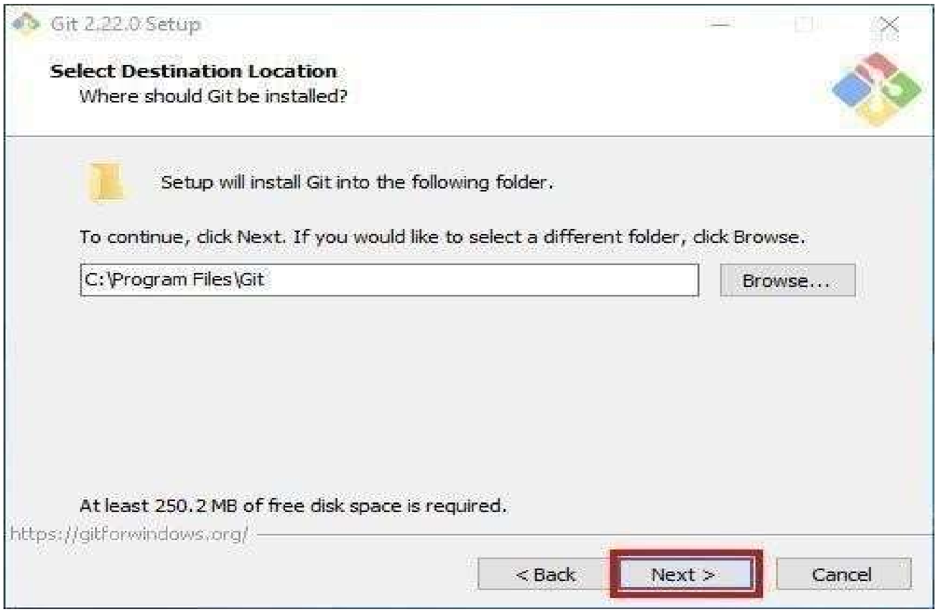
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

Click “Next” after you have read the license.

Step 4: Select Destination Location
Next, select the location you want to install Git Bash. I would recommend you just leave the default option as it is, and click “Next”.



Git 2.22.0 Setup

Select Destination Location
Where should Git be installed?

Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

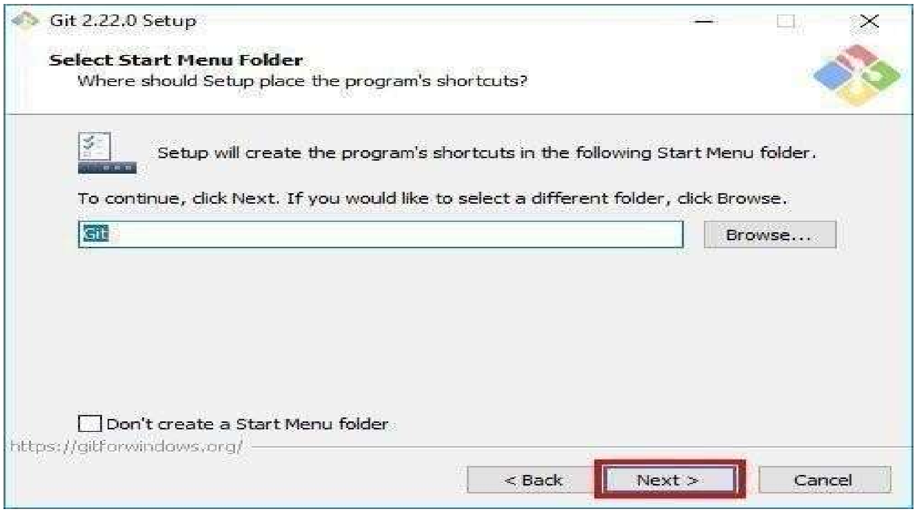
C:\Program Files\Git

At least 250.2 MB of free disk space is required.

https://gitforwindows.org/

Step 5: Select Components
Choose the components you want to install, or you can just proceed with the default options and click "Next". I prefer selecting the "Additional icons" component which creates a Git Bash shortcut on the desktop.

Step 6: Select Start Menu Folder
You can change the name of start menu folder here if you want, or just leave the default name and click "Next".



Git 2.22.0 Setup

Select Start Menu Folder
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Git

□ Don't create a Start Menu folder

https://gitforwindows.org/

Step 7: Choose the Default Editor used by Git
Next, select the default editor for Git to use. Choose the one you like and click "Next". I would recommend you proceed with **Nano** or **Notepad++**. Don't proceed with the default option "Vim" as ithas a steep learning curve.

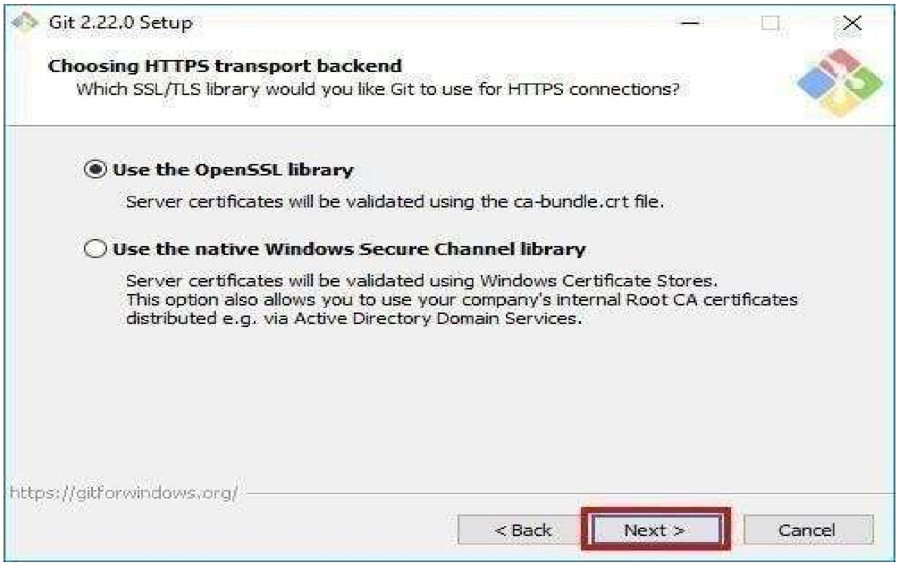
Step 8: Adjust your PATH Environment
Choose the option you want depending on where you want to use Git and click "Next".

Select **"Use Git from Git Bash only"** option if want to run Git and Bash commands from Git Bash only. This means that you won't be able to run Git commands such as git status on Windows CommandPrompt or Power shell. They will only be found on Git Bash.

Select **"Git from the command line and also from 3rd-party software"** option if you want to run Git commands on Windows Command Prompt or Power shell.

Select **"Use Git and optional Unix tools from the Command Prompt"** option if you want to use both Git and Bash commands on Windows Command Prompt or Power shell. This option will override some default Windows Command Prompt tools like find and sort. I don't use CMD or Power shell that much to worry about that. So, I will go ahead with this option by clicking "Next".

Step 9: Choose HTTPS Transport Backend
Next, select "Use the OpenSSL library" and click "Next".



Git 2.22.0 Setup

Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?

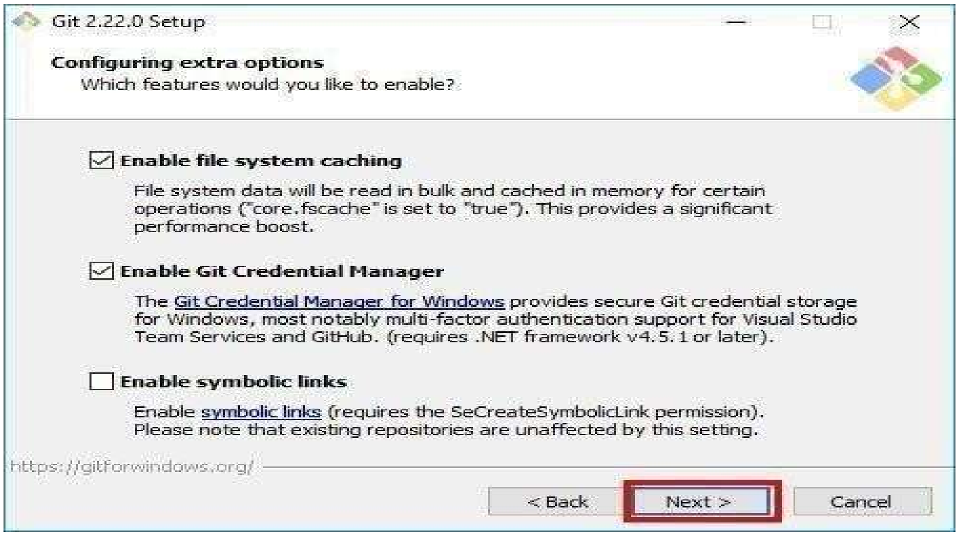
● **Use the OpenSSL library**
Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

Step 10: Configure the Line Ending Conversions
Select how Git should treat line endings in text files. It's probably safe to go with the default option "Checkout Windows-Style, commit Unix-style line endings". Click "Next" to proceed.

Step 11: Configure the Terminal Emulator to use with Git Bsh
Next, select the terminal emulator you want Git Bash to use. I will proceed with the default option "Use MinTTY(the default terminal of MSYS2) and click "Next".

Step 12: Configuring Extra Options
Select the features you want(the default options are fine) and click "Next".



Git 2.22.0 Setup

Configuring extra options

Which features would you like to enable?

☑ Enable file system caching

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☑ Enable Git Credential Manager

The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or later).

□ Enable symbolic links

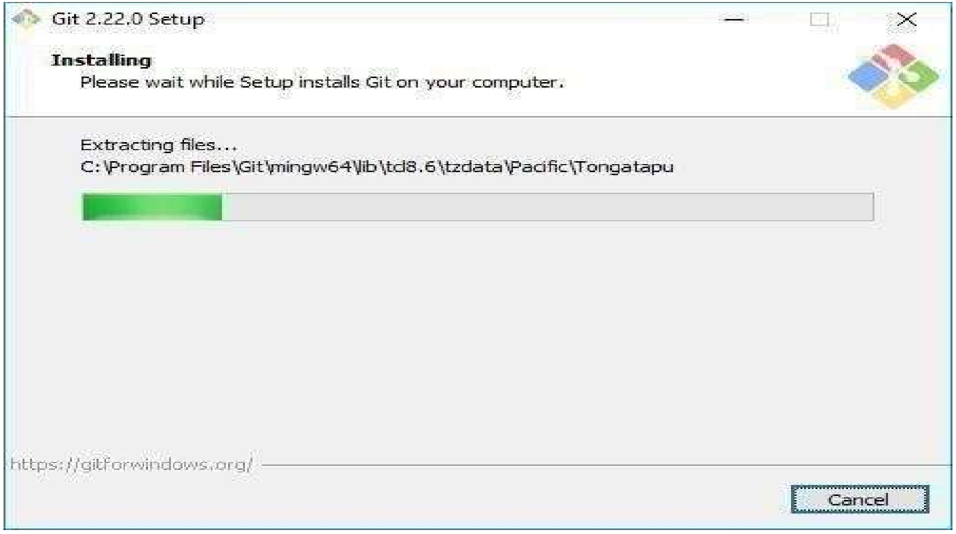
Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

Step 13: Configuring Extra Options

Enable experimental options if you want. Enabling them allows you to try out newer features that are still in development. I don’t enable this, so I will just proceed by clicking “Install” to start the installation process.

Step 14: Wait for Installation

Now, wait for a few minutes as the Setup Wizard installs Git on your computer.



Git 2.22.0 Setup

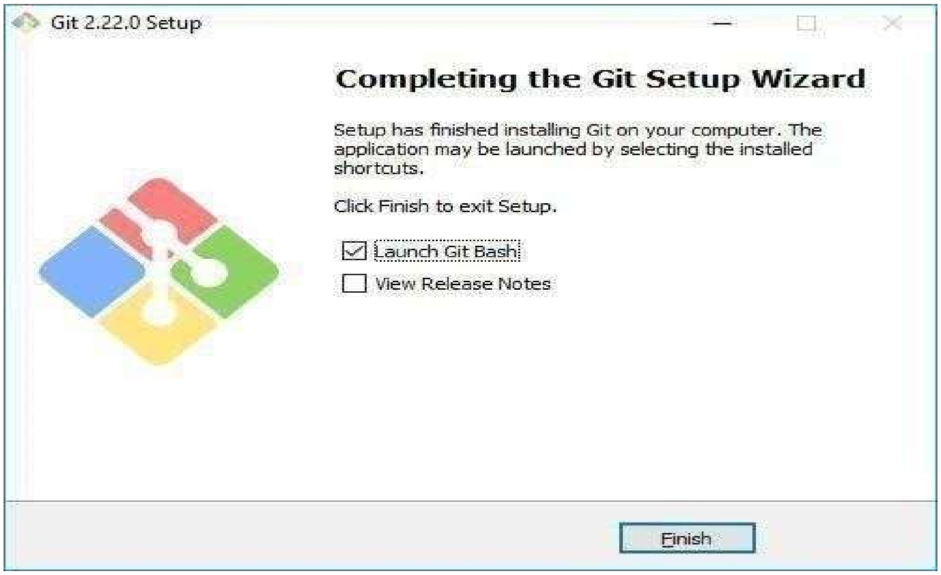
Installing

Please wait while Setup installs Git on your computer.

```
Extracting files...
C:\Program Files\Git\mingw64\lib\tcl8.6\tzdata\Pacific\Tongatapu
```

Step 15: Complete the Git Setup Wizard

After the installation has finished, check the “Launch Git Bash” and click “Finish” to launch Git Bash.



Git 2.22.0 Setup

Completing the Git Setup Wizard

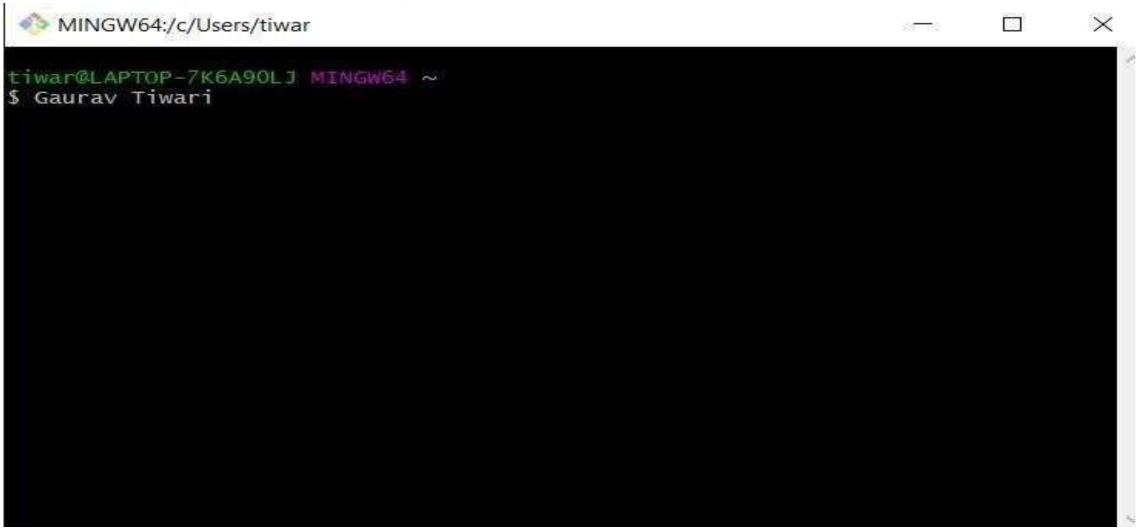
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch Git Bash

□ View Release Notes

The Git Bash terminal will now open and you will be able to enter Git and Bash commands.



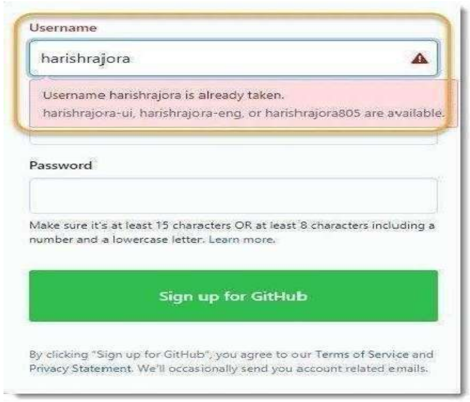
MINGW64:/c/Users/tiwar

```
tiwar@LAPTOP-7K6A90LJ MINGW64 ~
$ Gaurav Tiwari
```

Congratulations on successfully installing Git Bash.

Setting Up GitHub Account:

Setting your GitHub account is easy and very simple. To set the account visit **GitHub official** website. The login form will appear on the same page. Fill out the form with your details to create an account on GitHub.



Username

harishrajora

Username harishrajora is already taken.
harishrajora-ui, harishrajora-eng, or harishrajora805 are available.

Password

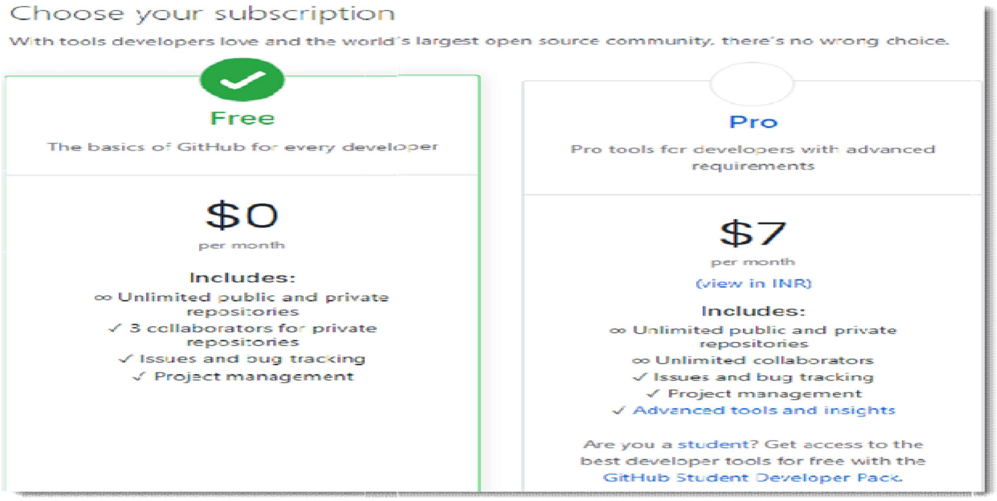
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

By clicking "Sign up for GitHub", you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account related emails.

Once you press **Sign up for GitHub** button, you will be prompted to verify that you are not a robot.

Choosing a GitHub Account Plan

Once you have verified your identity, you can choose the GitHub plan you want to subscribe for.



Choose your subscription

With tools developers love and the world's largest open source community, there's no wrong choice.

Free

The basics of GitHub for every developer

$0
per month

Includes:
∞ Unlimited public and private repositories
✓ 3 collaborators for private repositories
✓ Issues and bug tracking
✓ Project management

Pro

Pro tools for developers with advanced requirements

$7
per month
(view in INR)

Includes:
∞ Unlimited public and private repositories
∞ Unlimited collaborators
✓ Issues and bug tracking
✓ Project management
✓ Advanced tools and insights

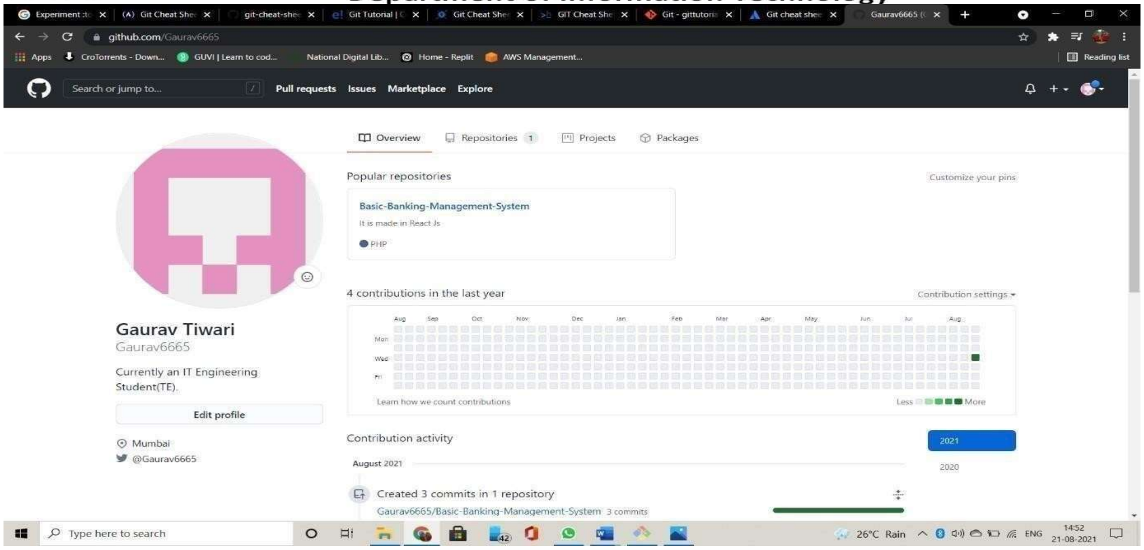
Are you a student? Get access to the best developer tools for free with the GitHub Student Developer Pack.

For this tutorial and in general as a beginner, **GitHub Free** plan is more than enough.

As the next step, you would be asked to verify your email address. You can verify it by clicking the link GitHub sent you on your email.

GitHub Account Dashboard

Now that the GitHub account is all set up, you can log in through your credentials on the **GitHub's website**. Logging in will land on the GitHub dashboard which is personalized for everyone according to interests.



Gaurav Tiwari
Gaurav6665

Currently an IT Engineering Student(TE).

Mumbai
@Gaurav6665

Popular repositories

Basic-Banking-Management-System
It is made in React Js
PHP

4 contributions in the last year

Contribution activity

August 2021

Created 3 commits in 1 repository
Gaurav6665/Basic-Banking-Management-System 3 commits

Conclusion: Hence, we have Understood Version Control System/Source Code Management and install git and create a GitHub account.