**Git & GitHub**

* **What is Git?**
  + Git is a Distributed Version Control System (DVCS).
  + Git is nothing but a system which stores history of files.
  + Git supports branching concept in project development.
  + Git allows multiple users to work together on a same project without any mess of code.
  + Git & GitHub are not the same.

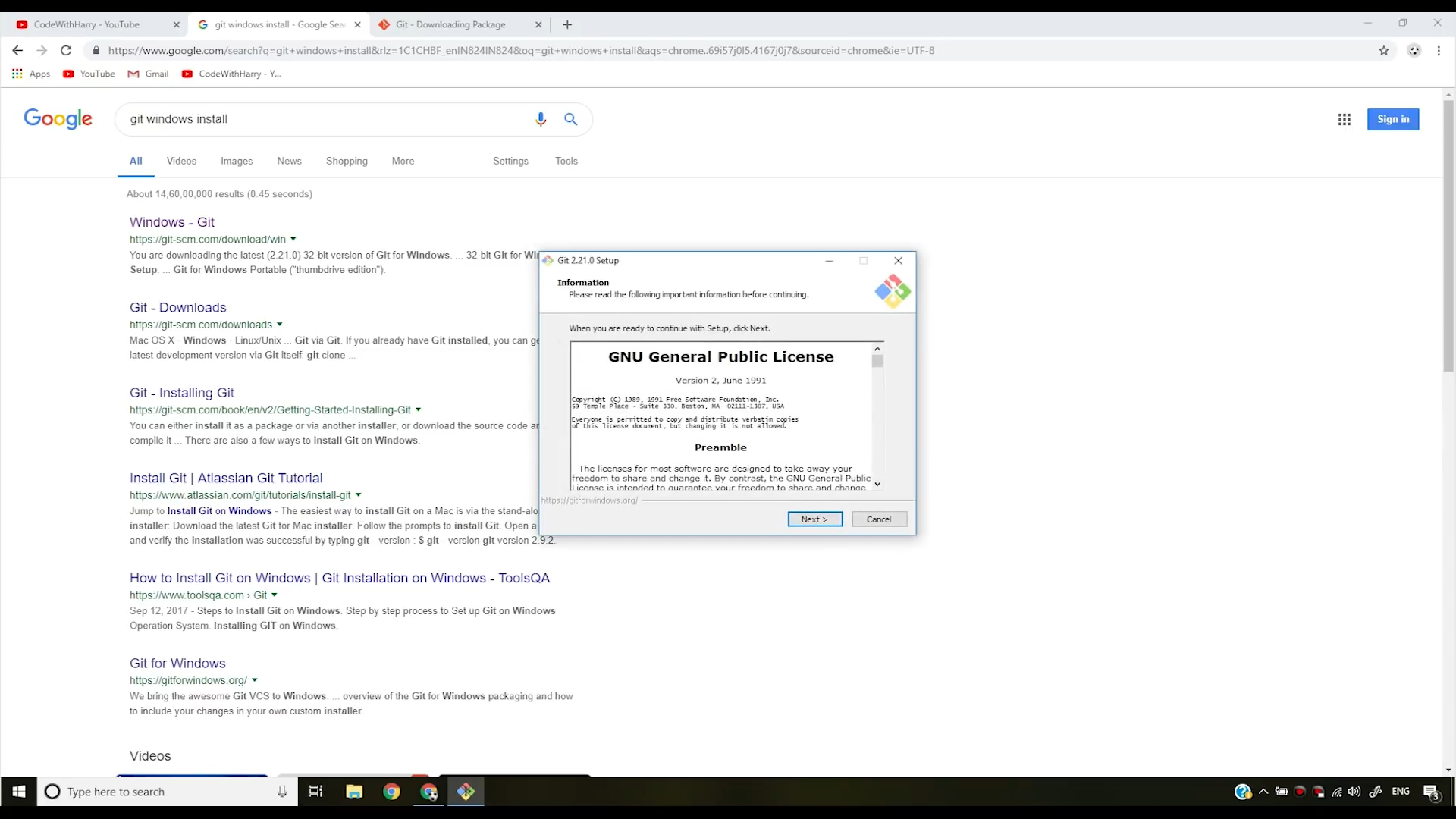
**Website : -** [www.git-scm.com](http://www.git-scm.com)

* **What is GitHub?**
  + GitHub is a web-based version-control and collaboration platform for software developers.
  + GitHub facilitates social coding by providing a web interface to the Git code repository and management tools for collaboration.
  + GitHub can be thought of as a serious social networking site for software developers.
  + You can host your code on remote repository
  + You can also create public as well as private repository

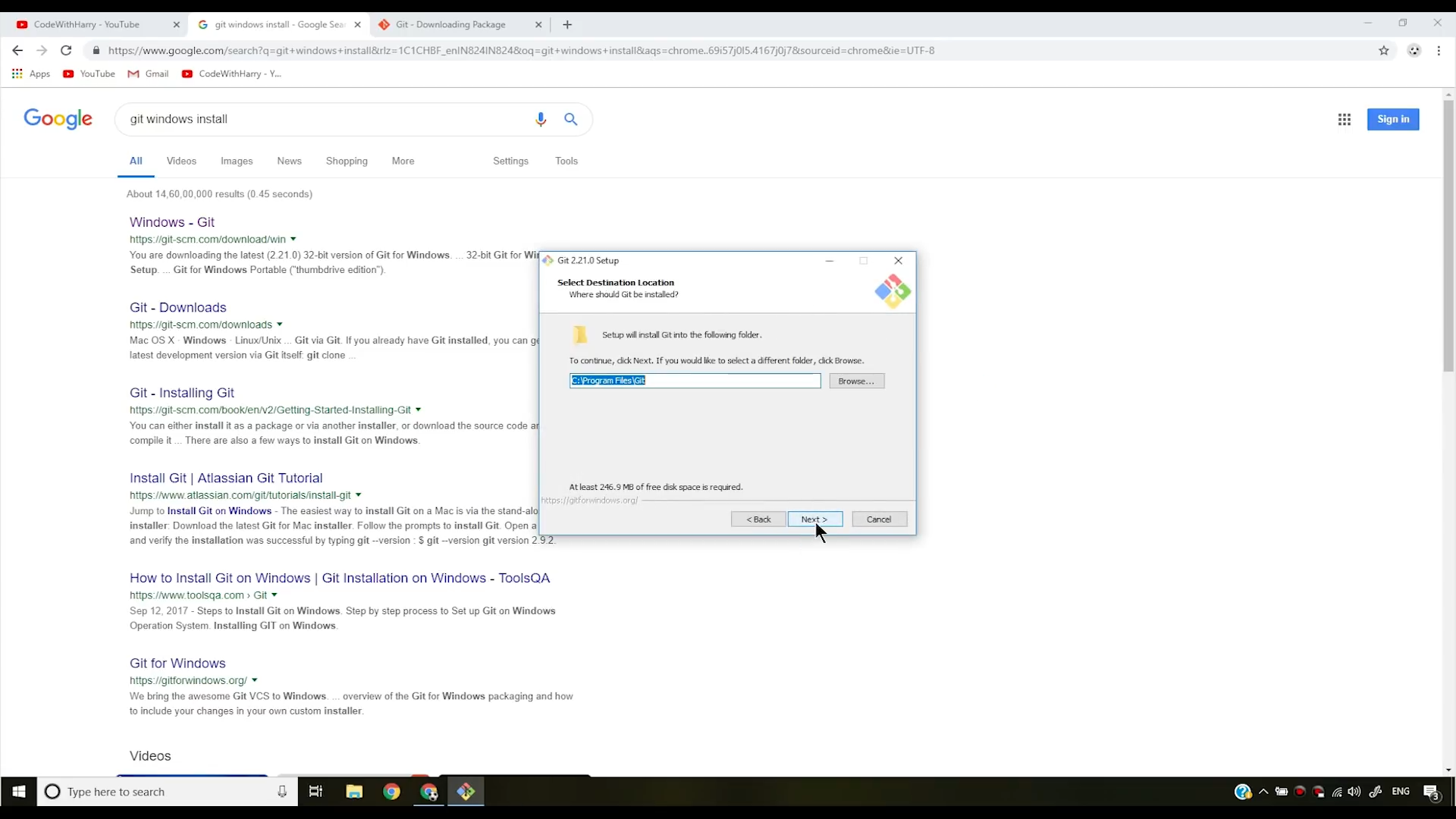
**[Website : -](https://github.com)**<https://github.com>

* **Installing Git**

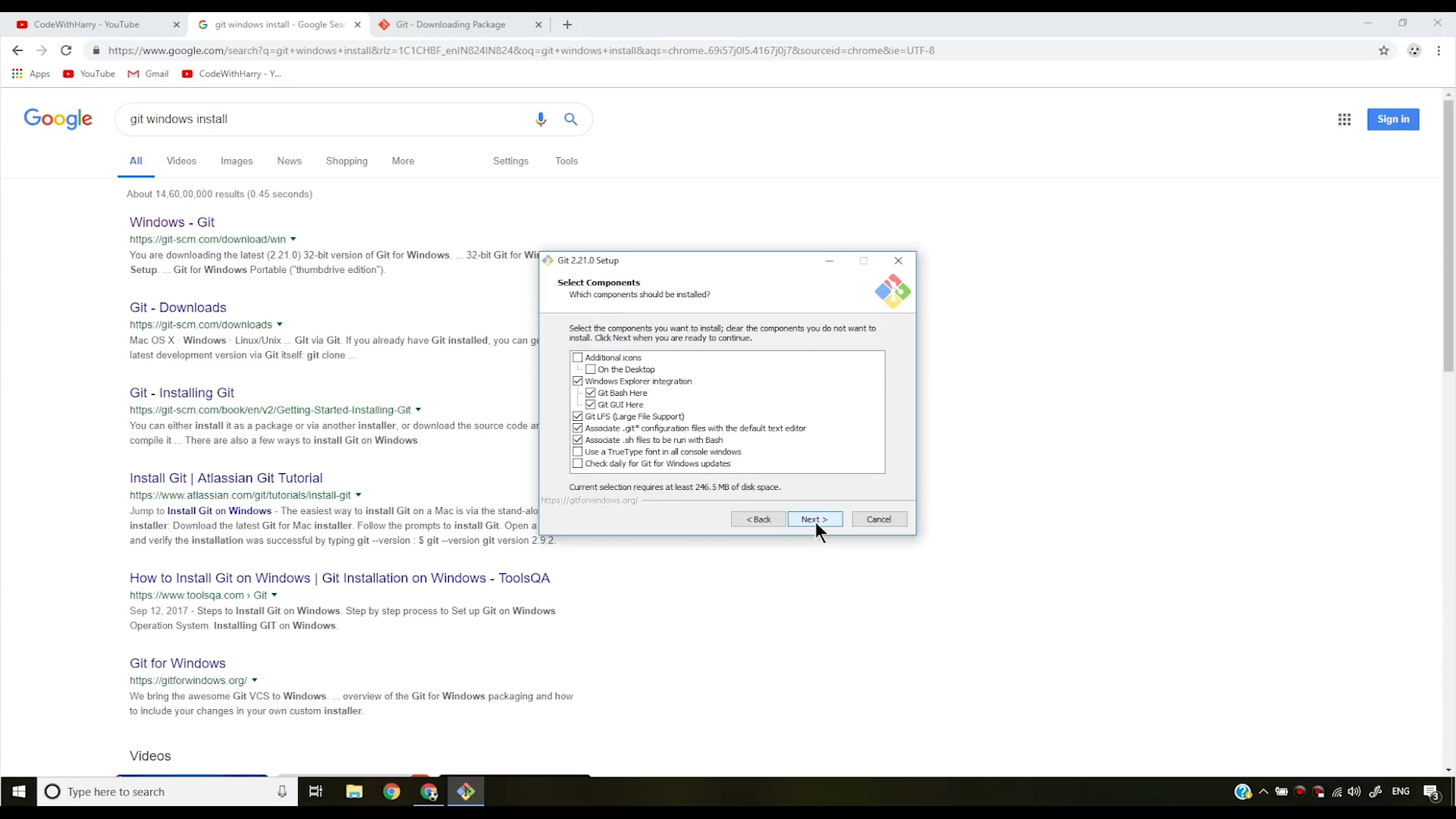
1. Download the setup from the website given above.
2. Run the Setup.
3. Click on Next Button.



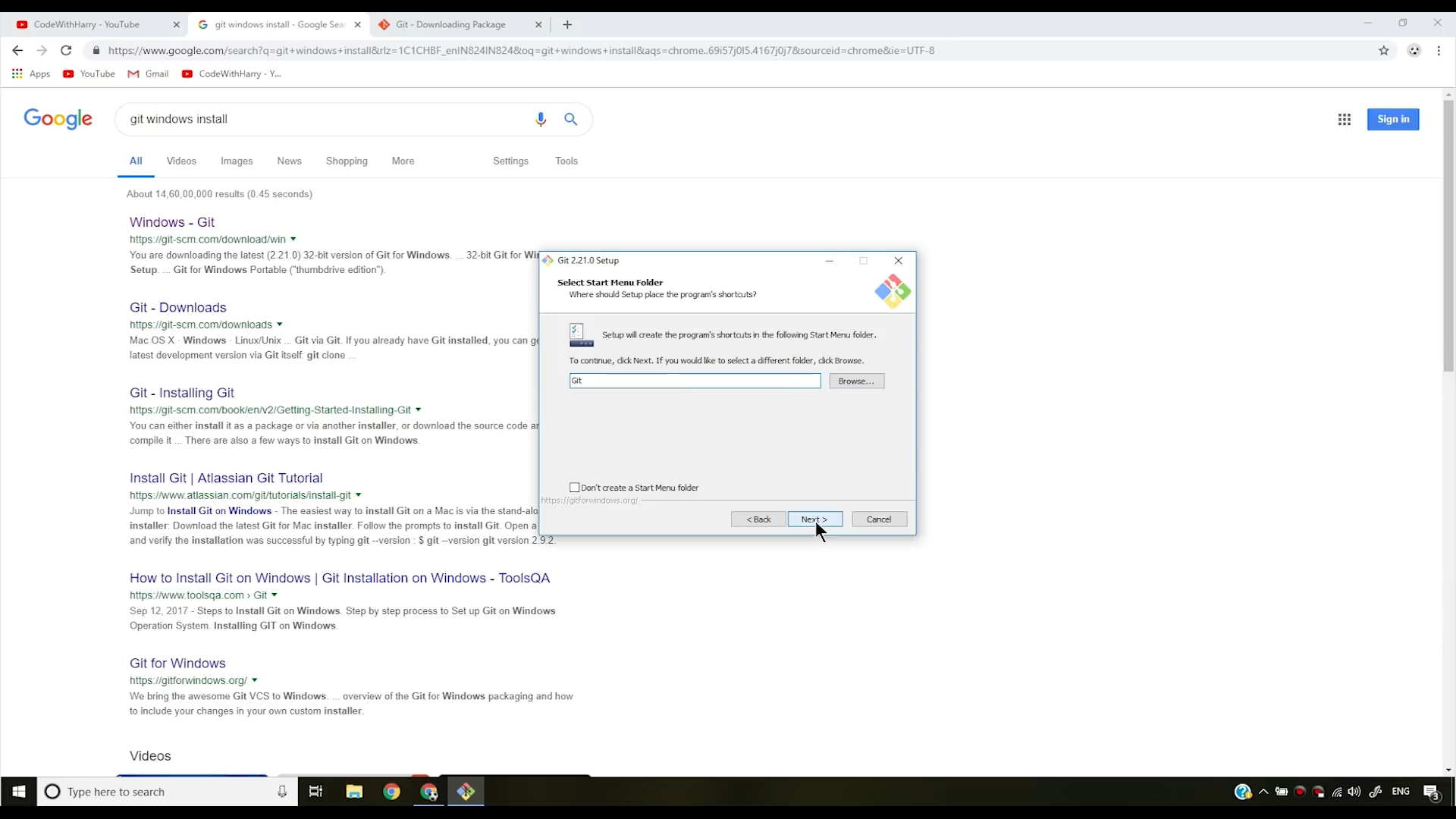
1. Select the folder where you want to install the git.



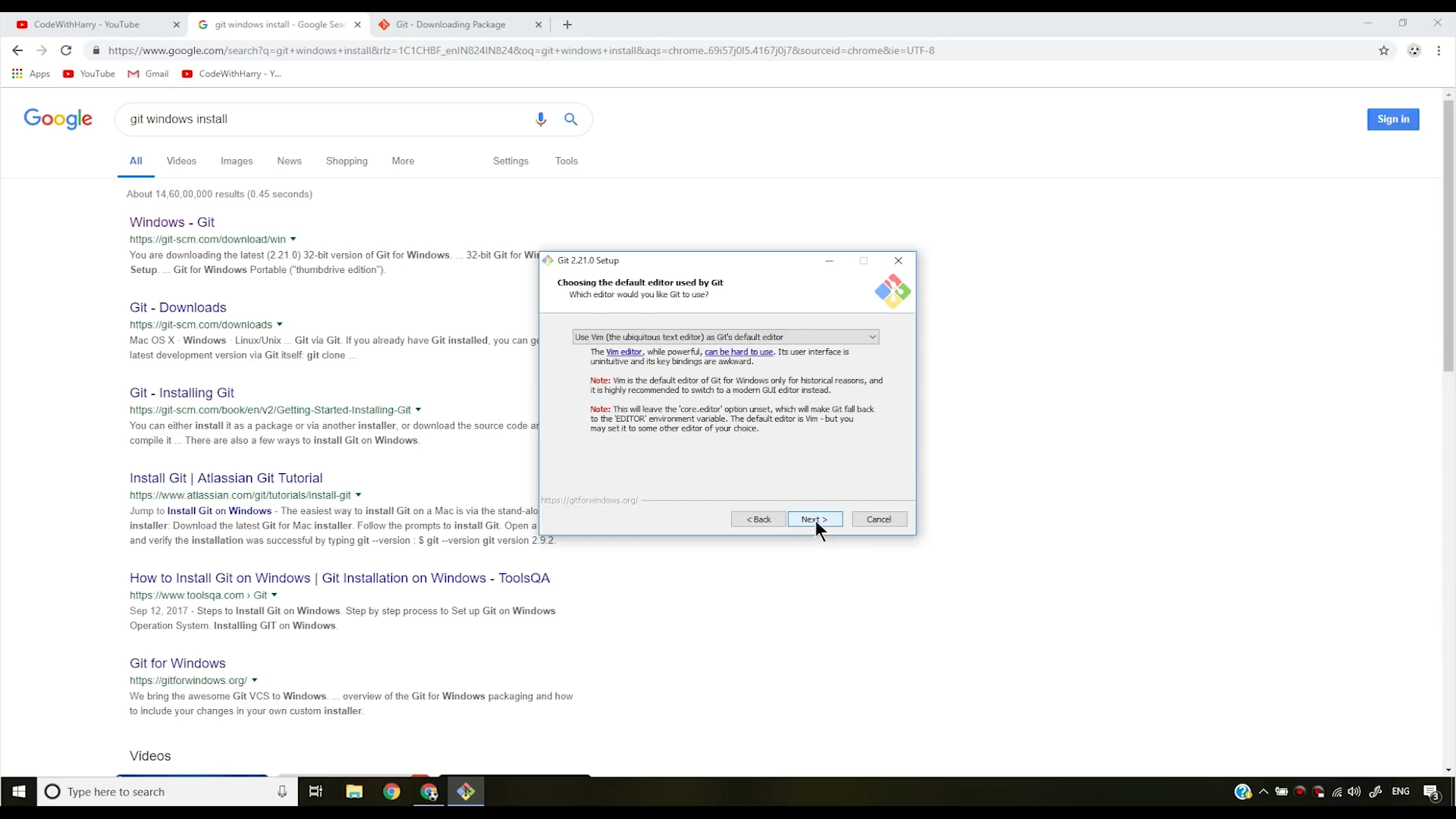
1. Check the appropriate checkbox and click Next.



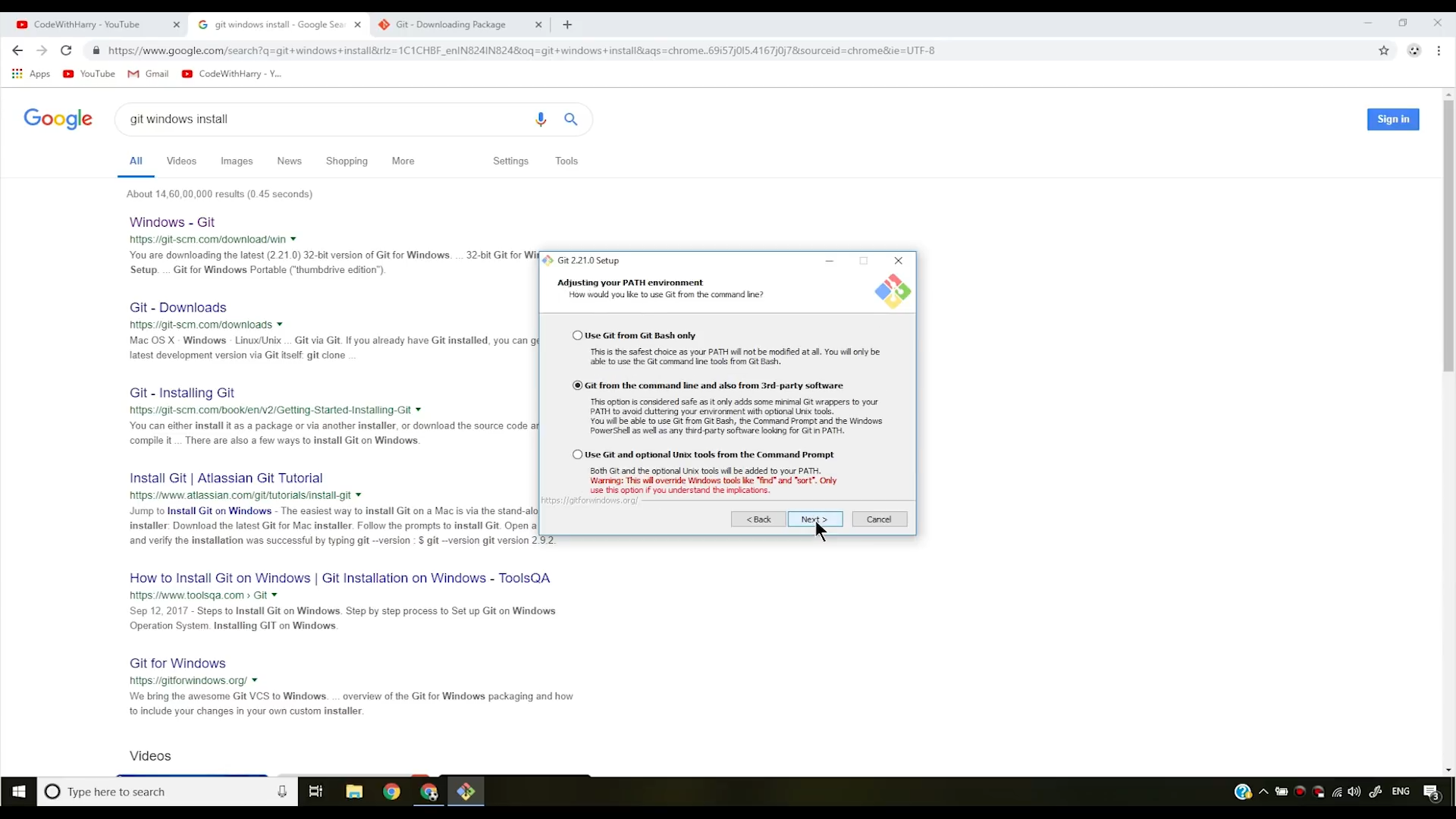
1. Select the start menu folder and click Next.



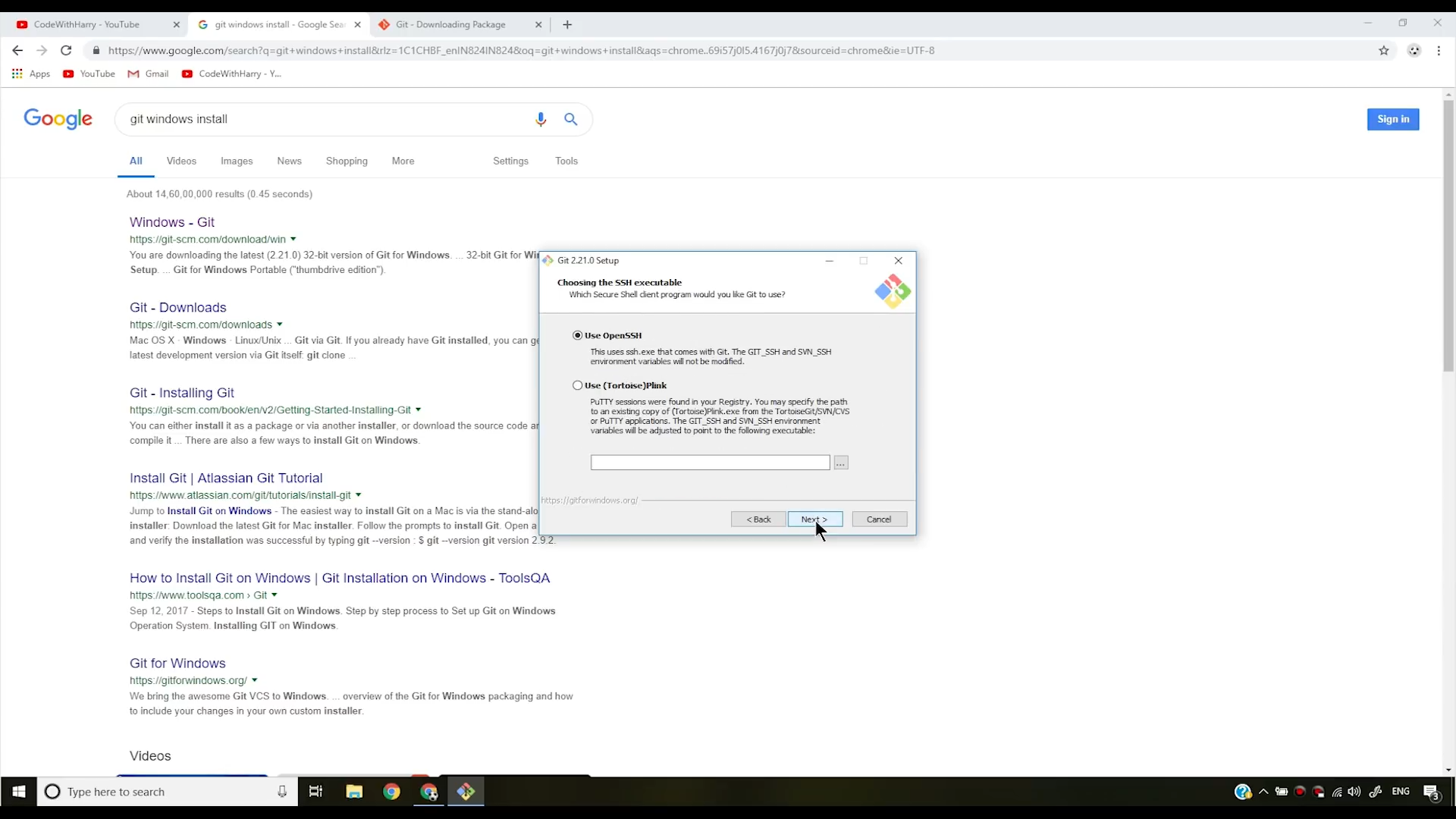
1. Choose the default editor and click “Next”.



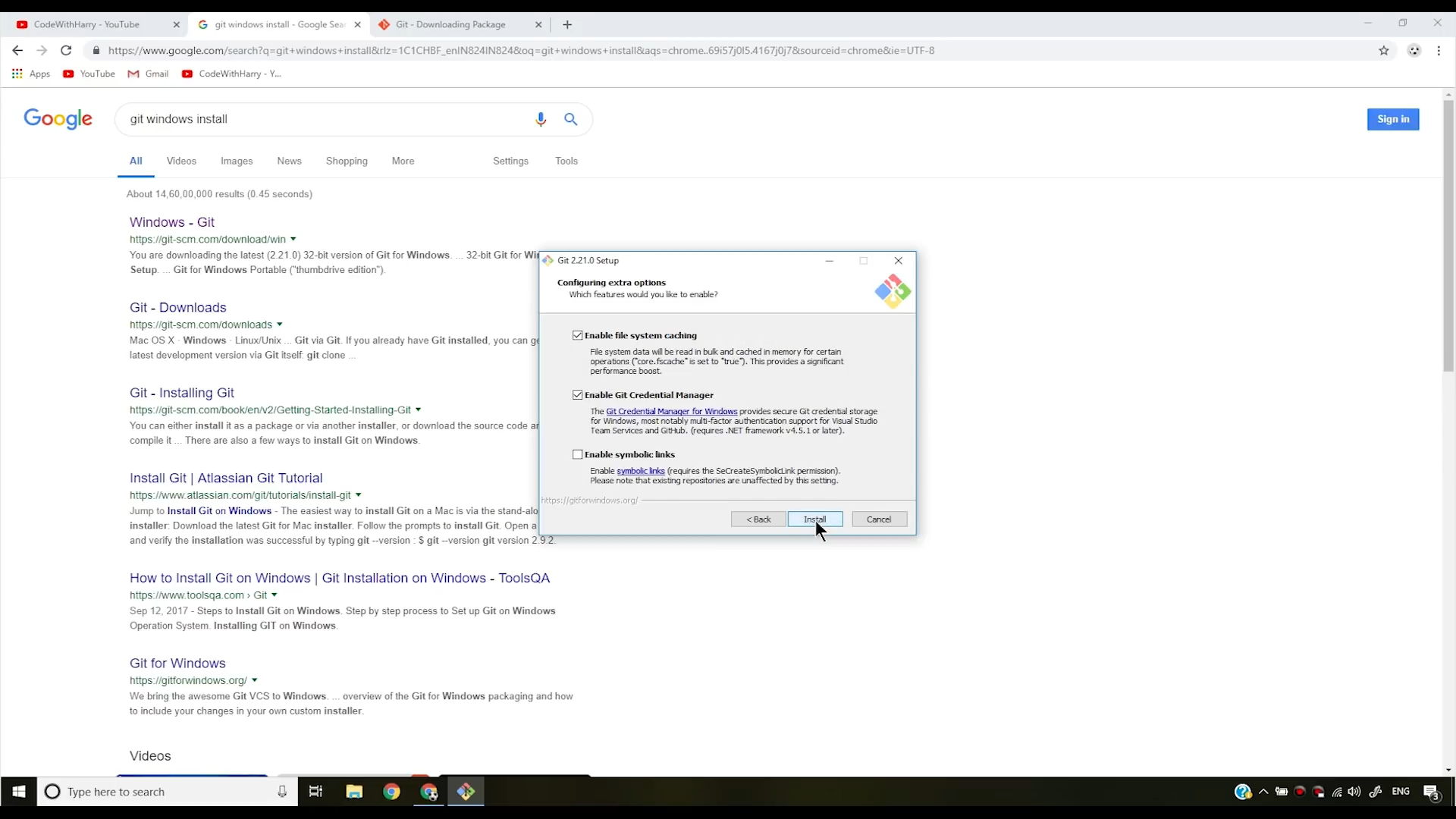
1. Set the path and click “Next”.



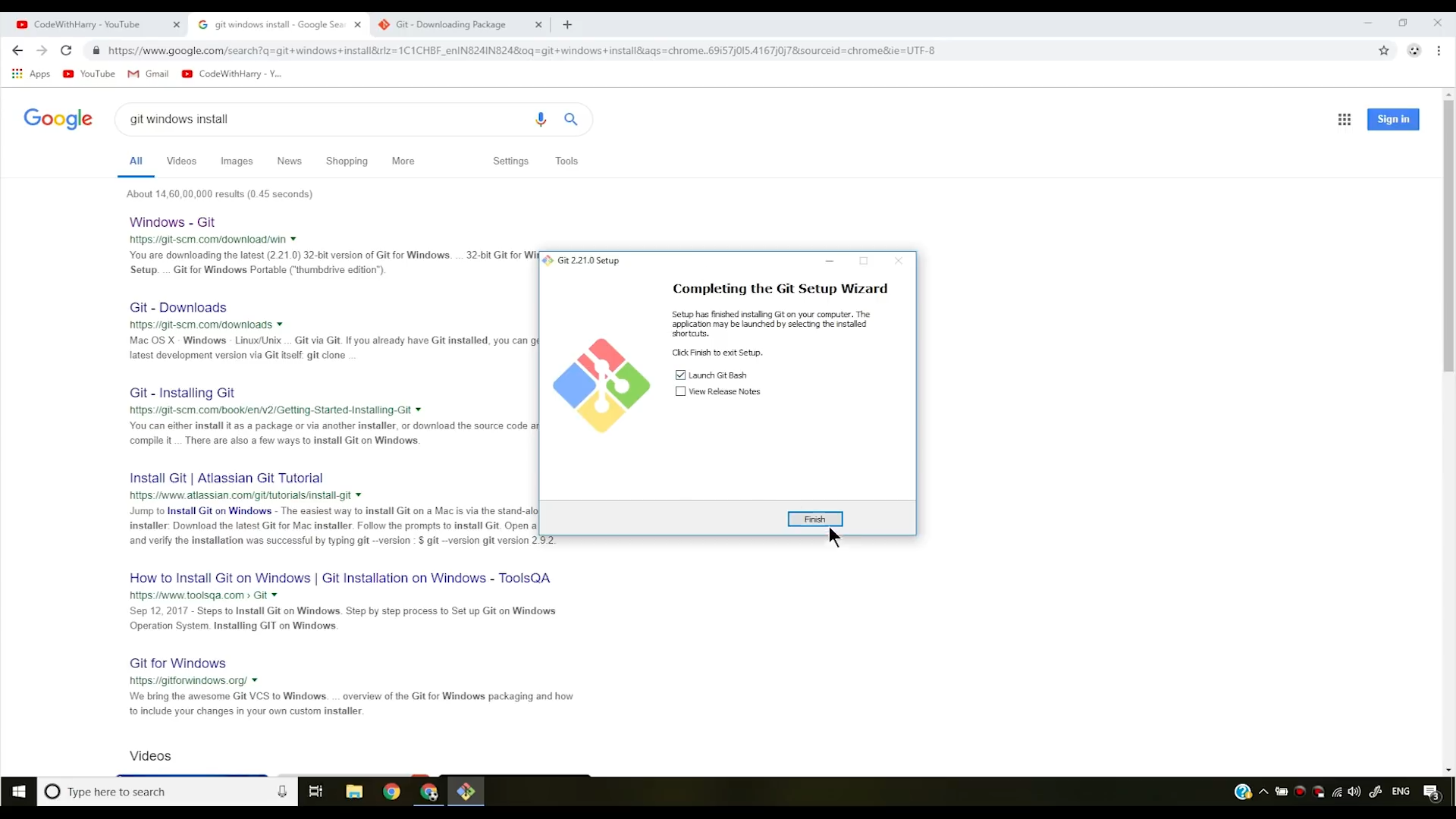
1. Choose the SSH executable and click “Next”. You can also set this after finishing the installation.



1. Configure the extra options and simply click on the “Install” button to begin the Installation.



1. Click the “Finish” button after the installation. You can check the checkbox to launch the “Git Bash”.



**Here are some commands which will be helpful to operate with it Bash (Git Terminal)**

* ls
  + List all the files and folders in present working directory.
* clean
  + clear the terminal
* ls -lart
  + display all hidden files and folders
* pwd
  + Shows the path of present working directory
* cd <directory\_name>
  + change the current directory to the specified directory.
  + E.g
    - cd testgit
* cd ..
  + change the present working directory to the parent directory
* git --version
  + display the current version of the git install on the system.
* git config –global user.name <user name>
  + change the name in Git system.
  + E.g
    - git config –global user.name Riyaz
* git config –global user.email <user email id>
  + change the email in Git system.
  + E.g
    - git config –global user.email [xyz@gmail.com](mailto:xyz@gmail.com)
* git config –global user.name
  + display username
* git config –global user.email
  + display email id
* git config –global –edit
  + open the config file in vim editor
* mkdir <director\_name>
  + create a directory with particular name
  + e.g
    - mkdir testgit
* git init
  + initialize the git in current directory
* git status
  + display the current status of the git folser/repository
  + it display the information about files like..
    - committed file
    - modified files
    - untracked files etc…
* git status -s
  + overview of git status command
  + short hand for git status command
* git add <file\_name>
  + add the particular file to git stagging area.
  + E.g
    - git add Sum.java
* git add .
  + Add all the files to the git stagging area.
* Git add -A
  + Add all files in stagging area
* git commit -m “message”
  + Save the changes to the git database.
  + E.g
    - git commit -m “initial commit”
* git commit -a -m “message”
  + skip the staging area and direct commit the changes
  + e.g
    - git commit -a -m “skip the stageing and direct commit”
* git rm <filename>
  + remove the specified file from staging area and delete the file from hard disk as well
* git rm –cached <filename>
  + remove file from only staging area
* git log
  + display the list of all previous commits.
* git log -p <-number>
  + display only last specific commits
  + e.g
    - git log -p -1
* git diff
  + compare working directory with staging area and display the changes
* git diff --staged
  + compare staging area with last commit and display the changes
* git checkout <hash\_code / branch\_name>
  + toggle the head to the particular hash code or specified branch.
  + E.g
    - git checkout dev
    - git checkout master
    - git checkout dbbc18ff7a5201fba2fce90b7b1e75f713aaca70
* git checkout <filename>
  + modify the specified file with the last commit
* git checkout -f
  + modify all files according to last commit
* git branch
  + show all the branches
* git branch <branch\_name>
  + create a branch with specified name
  + E.g
    - git branch dev
* git checkout <branch\_name>
  + switch the head to the specified branch
  + E.g
    - Git checkout dev
* git checkout -b <branch\_name>
  + create a branch and checkout the same branch at a time
  + E.g
    - git checkout – b Riyaz/multiply
* git merge <branch\_name>
  + merge the specified branch in current branch
  + E.g
    - git merge Riyaz/multiply
* touch <file\_name>
  + create a blank/empty file with particular name
  + E.g
    - touch .gitignore
* git remote add <destination\_alias> <destination\_address>
  + create a alias of particular url
  + E.g
    - git remote add origin <https://github.com/786riyaz/testgit>
* git remote set-url <alias> <url>
  + modify the alias url
  + E.g
    - git remote set-url origin [git@git.com](mailto:git@git.com)

* git remote -v
  + display github fetch & pull links
* git branch -M master
* git push -u <destination> <branch\_name>
  + these both commands will push the code of the branch the GitHub database.
  + E.g
    - git push -u origin master
    - git push -u origin dev
* git clone <path\_of\_remote\_repository>
  + clone another repository to the current directory
  + E.g
    - git clone <https://github.com/testgit.git>
* git clone <path\_of\_the\_remote\_repository> <folder name>
  + clone another repository in specified directory
  + e.g
    - git clone <https://github.com/jaunchopanza/cppblog.git> Riyaz
* vim <file\_name>
  + open the particular file in vim editor.
  + E.g
    - Vim Readme.txt