GitHub, Student

By: Ebonie I. Gadson
Montgomery College, 2018

These Slides Will Answer the Following Questions:

What is GitHub?
How do I use GitHub?
How do I create Private Repository?
How do I use GitBash and desktop version?

What is GitHub?

GitHub is a *version control system that (VCS)* that lets users save their code online, track changes of their code (versions), and collaborate with others on a particular project. A version control system records changes to files over time and let's users review this changes later.

Git vs. GitHub

It was built off of **Git** which is also a VCS; however, Git only uses the command line tool when making changes to a project. **GitHub** allows users to make changes online, using the command tool (GitBash or the terminal), and in GitHub Desktop.

VS

GitHub Terms

Repository is where all files are stored for a particular project. Each project has its own repository where multiple versions and branches are created and tracked.

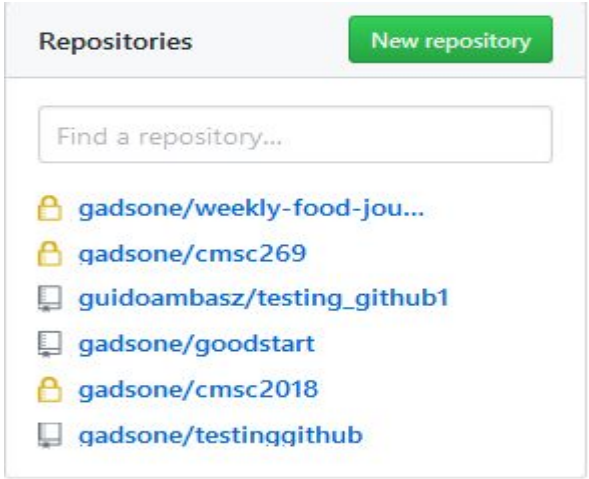
Branches are the versions of a user's code inside a repository that is created without altering original state of the project (usually the master branch).

Master branch is created once a repository is created and can store all versions of a project when branches merge into it.

Cloning allows for the online repository to be downloaded locally on a user's computer.

Public vs. Private Repositories

Public repositories are projects are visible to the entire GitHub community is great for collaborating with users all over the world. **Private repositories**, however, are only visible to the user that creates it unless they decide to add collaborators are on the project. Repositories that have lock represent private while the public repositories have an open source symbol.

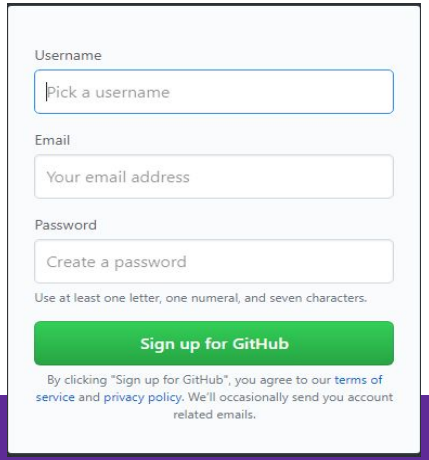


Create a GitHub Account

Create your GitHub account using your Montgomery College Email (MYMCID@montgomerycollege.edu):

https://github.com/

With a valid student email, you are able to create private repositories for free.



Request a discount

After signing up for github go to https://education.github.com to request a free 2 year discount. GitHub is a open source software that allows users to create multiple public repositories, but users need to request a discount for unlimited private repositories. This is free with student verification!

Follow the next steps to activate student development pack.

Real-world tools, engaged students

GitHub Education helps students, teachers, and schools access the tools and events they need to shape the next generation of software development.

GitHub Student Developer Pack

The best developer tools, free for students

GitHub Campus Experts

Training to enrich the technology community at your school

GitHub Field Day

Unconferences for leaders of technical student communities

GitHub Classroom

The GitHub workflow, scaled for the needs of students

GitHub Campus Advisors

Teacher training to master Git and GitHub

Learn to ship software like a pro.

There's no substitute for hands-on experience, but for most students, real world tools can be cost prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.

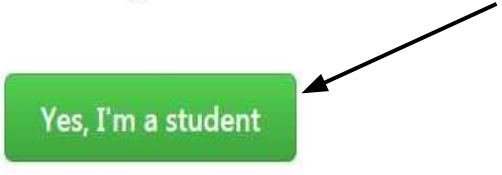
Get your Pack

Are you a student?

The GitHub Student Developer Pack is only available to students aged 13 or older.

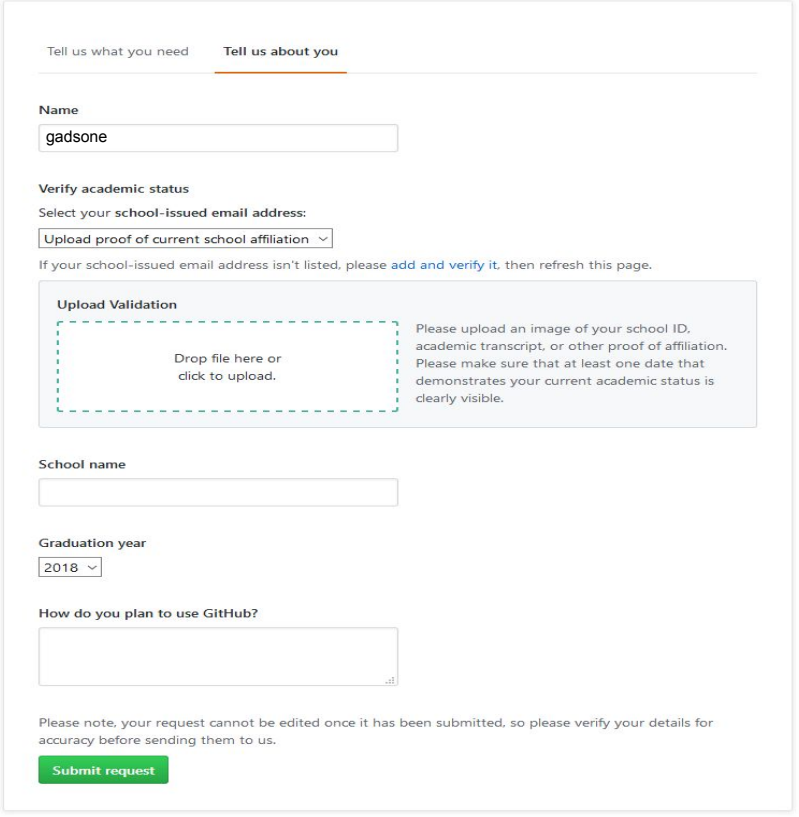
Before you receive access to the offers we need to verify that you are a student.

Teachers, researchers, faculty, staff, and other educational users can get free and discounted access to GitHub, but are not eligible for the Pack. If you're not a student, you can still request a regular GitHub for education discount.

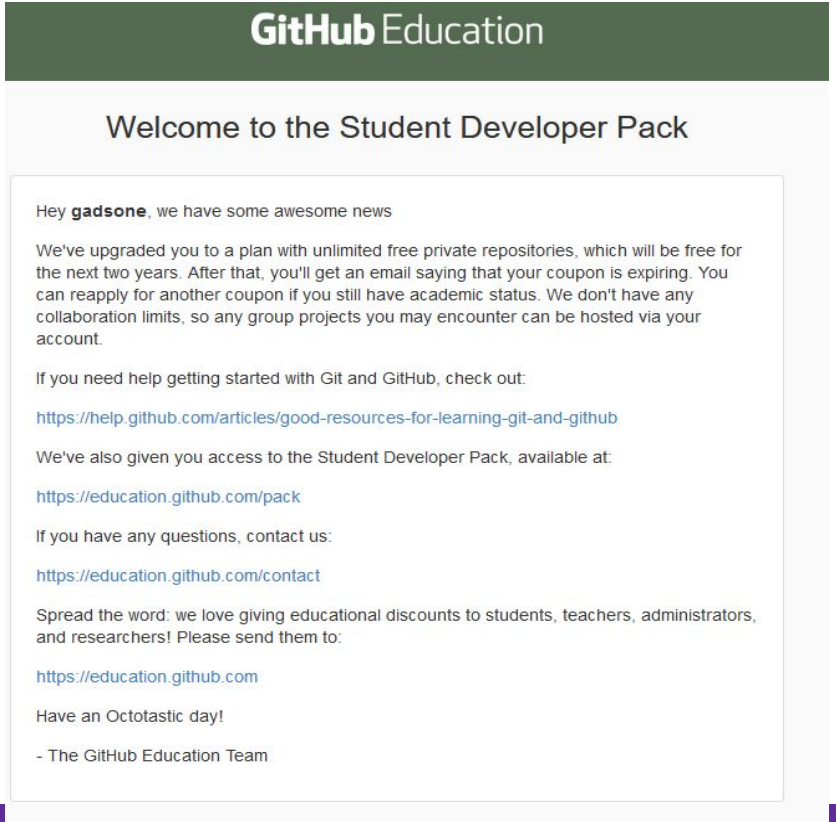


No, I'm not a student but would still like a discount

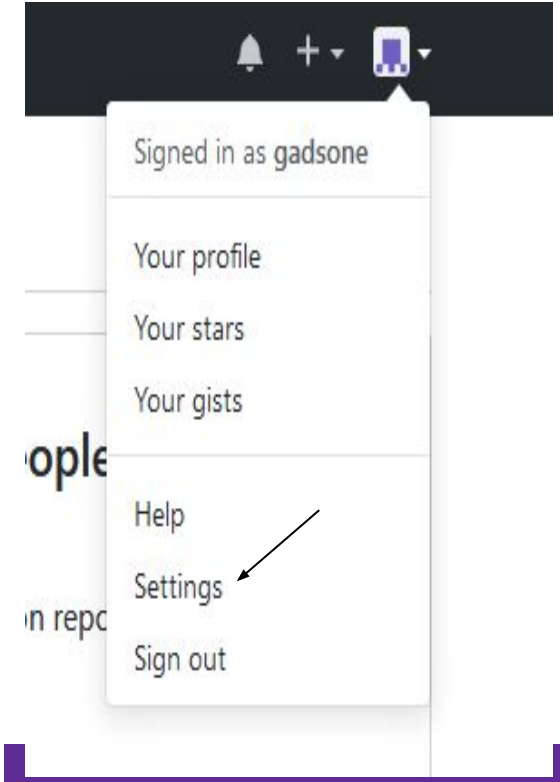
Fill in the information as seen and submit your request.

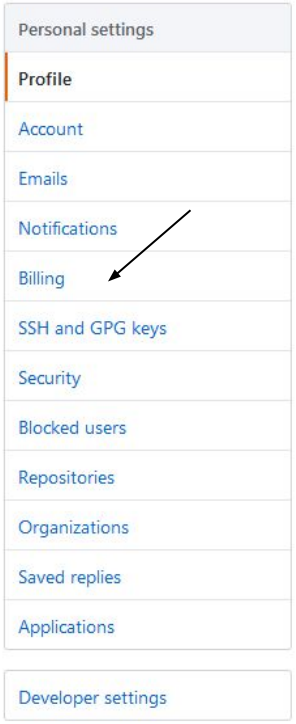


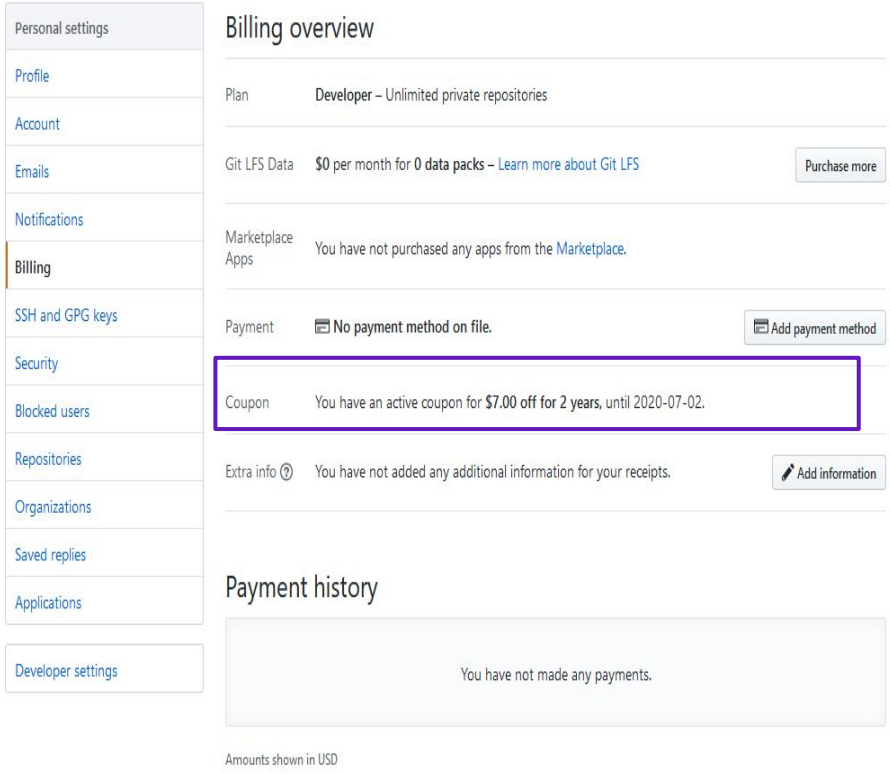
Check college email to see when you have leveled up to student pack.



After submitting your request go to the billings tab on github and you should see your coupon.





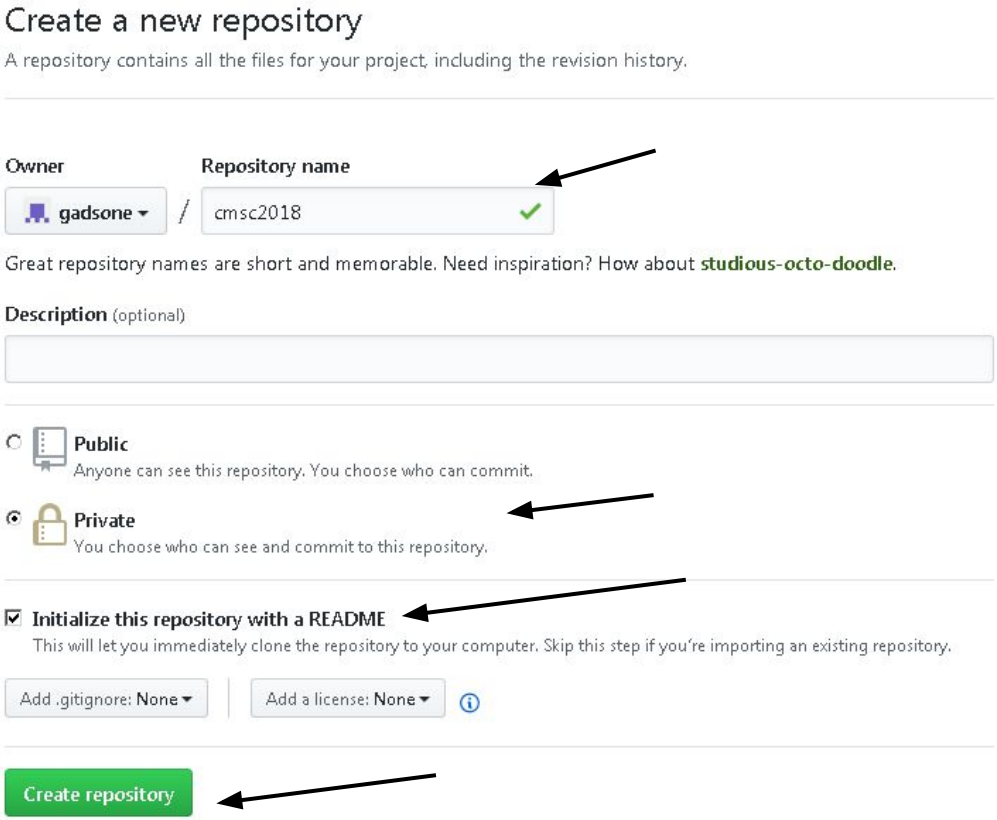


Create a Repository

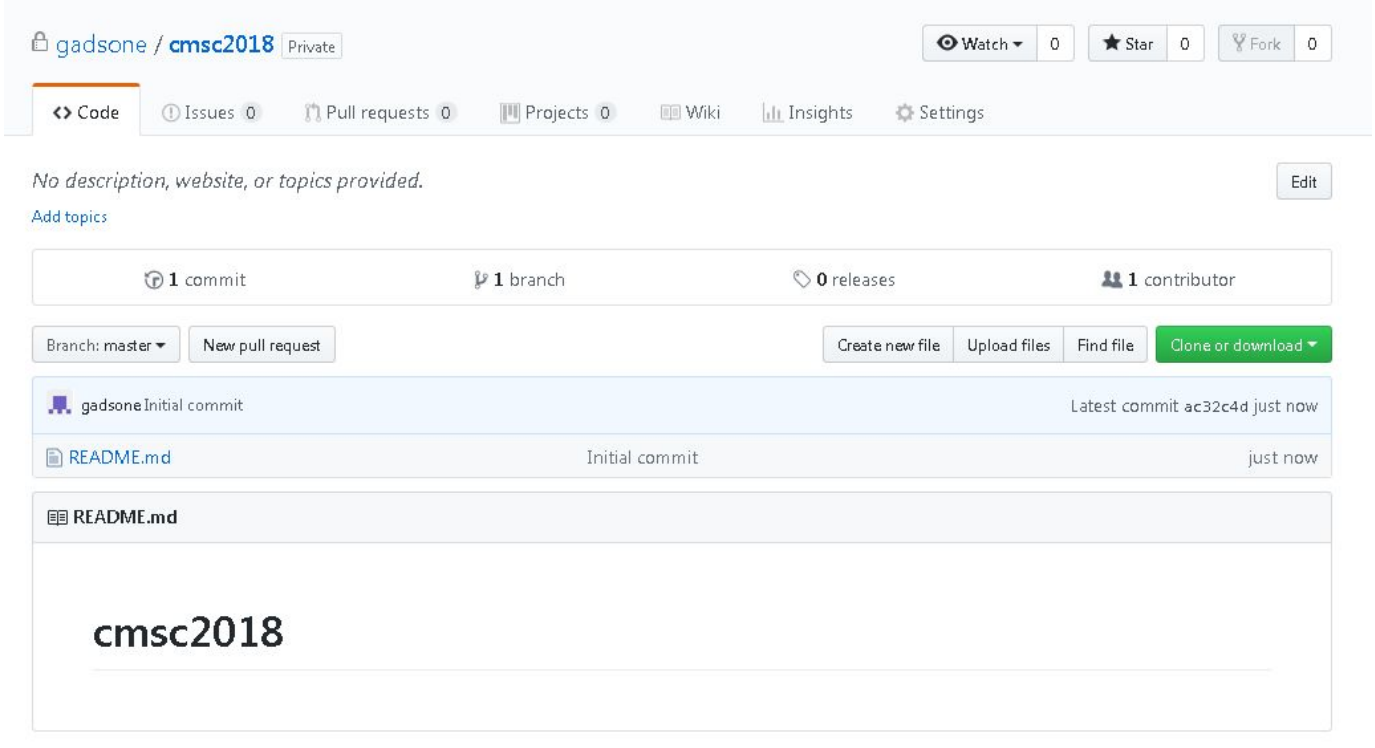
- Click on Start a Project to create a repository.

- After naming your repository, choose the private option for your repository.

Name repository and select either public or private. Next, check initialize README box. Then, click “create repository”



You have successfully created a private repository. You can now add files and invite others.

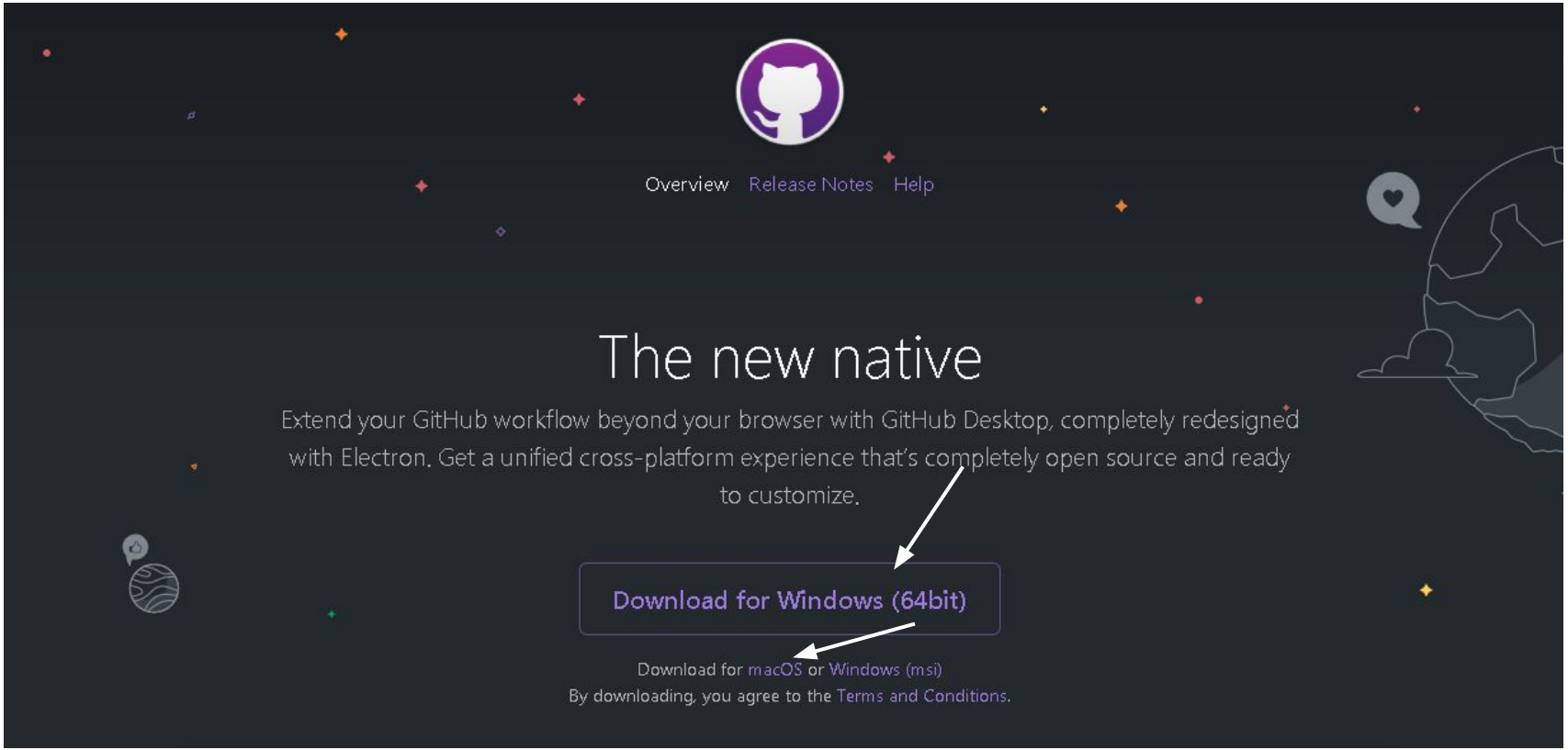


Download GitHub Desktop

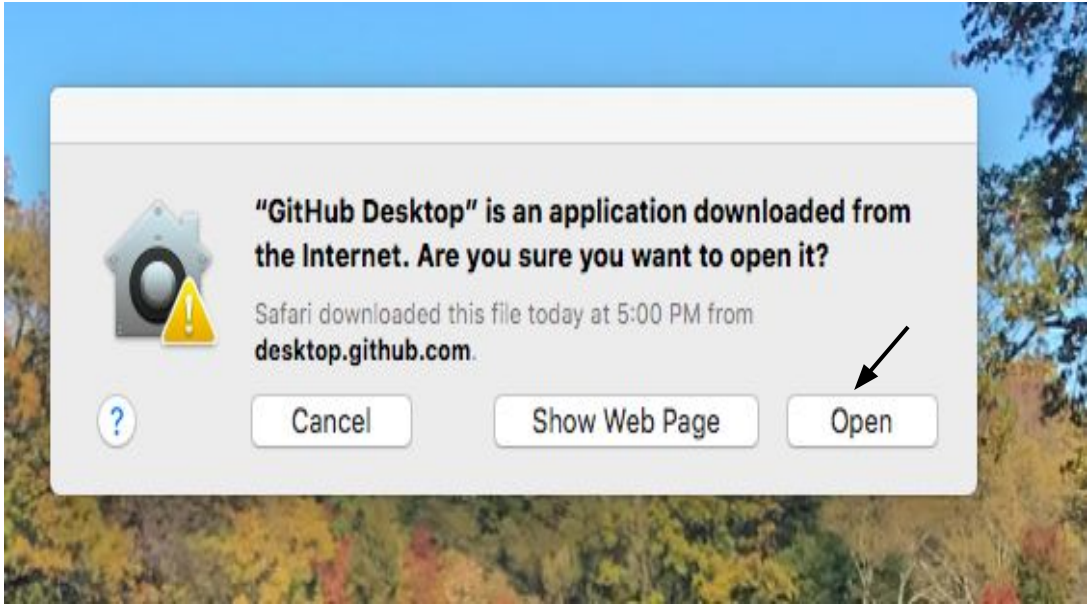
Download GitHub desktop to manage your workflow for Windows and macOS.
https://desktop.github.com/

Then complete the following installation steps:

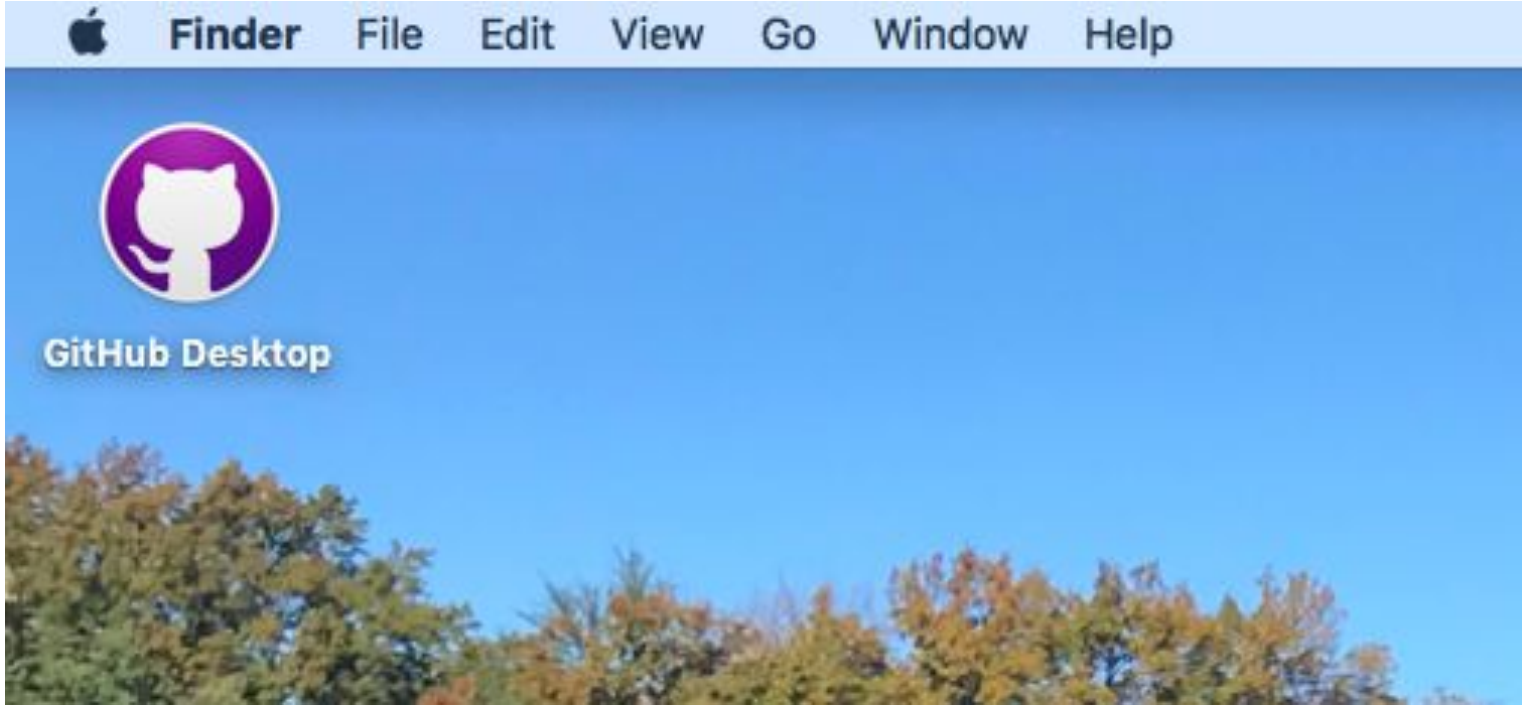
Download the 64-bit version for Windows or macOS.



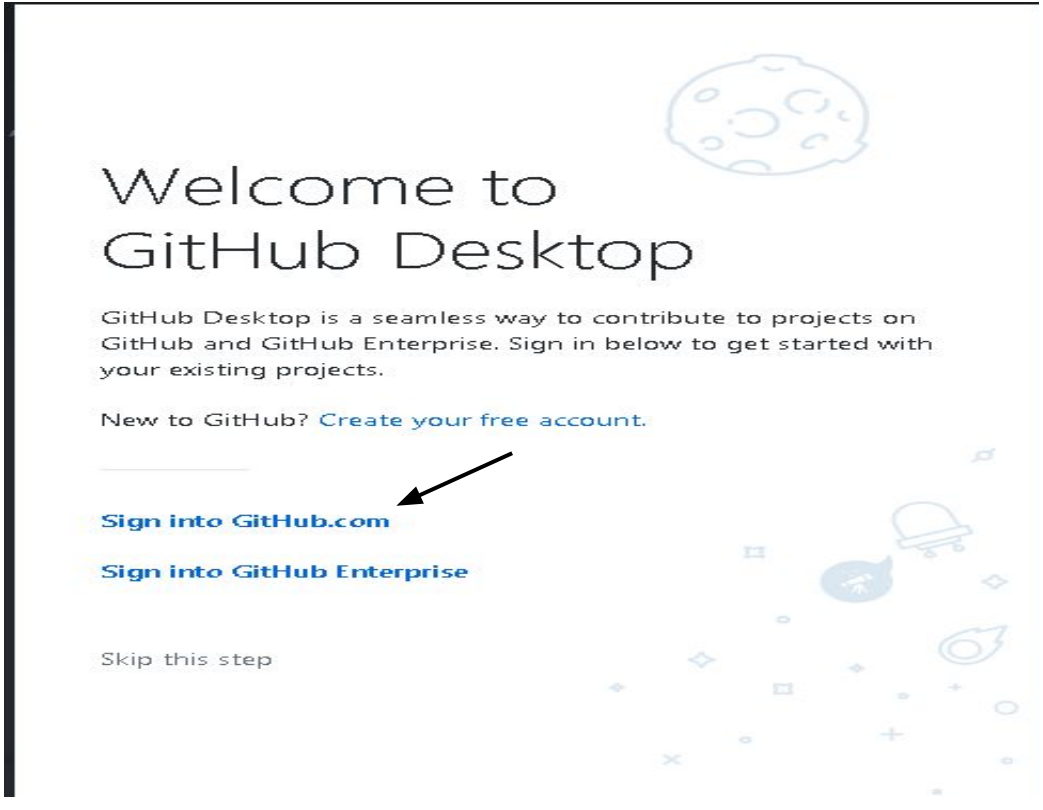
Run GitHub Desktop for Windows or macOS



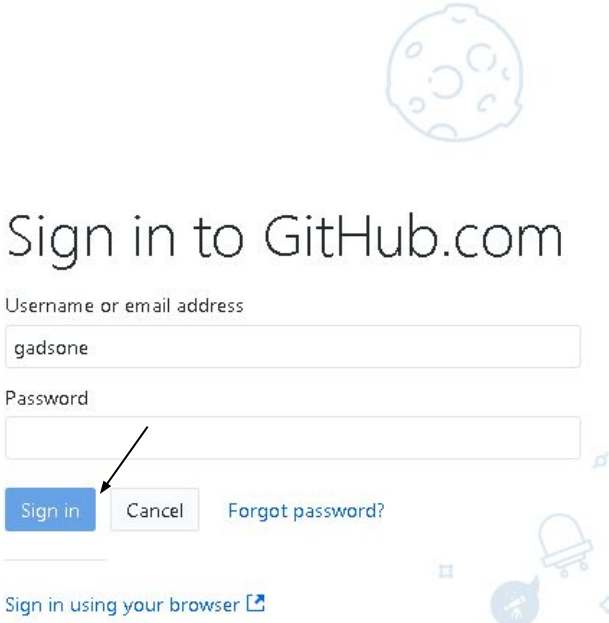
In macOS, open the GitHub desktop application.

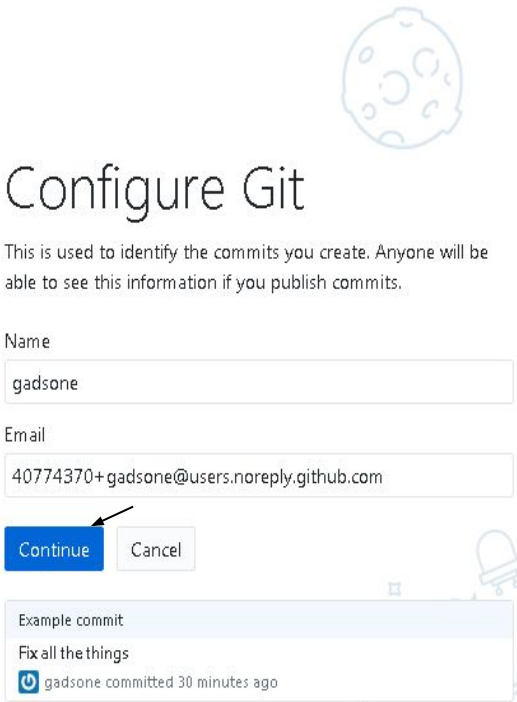


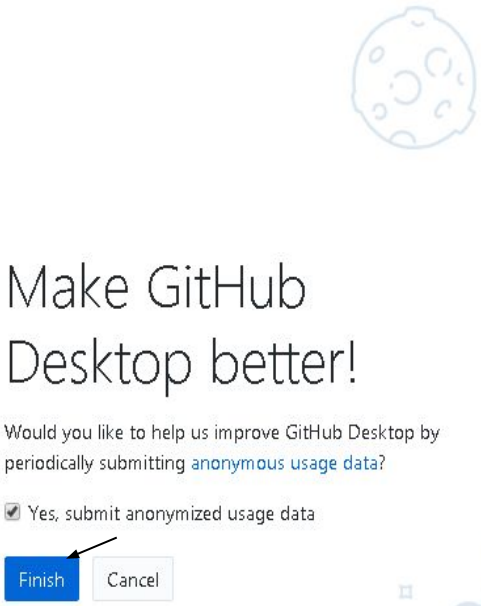
Select “Sign into GitHub.com” for both Windows and macOS



Login to GitHub Desktop

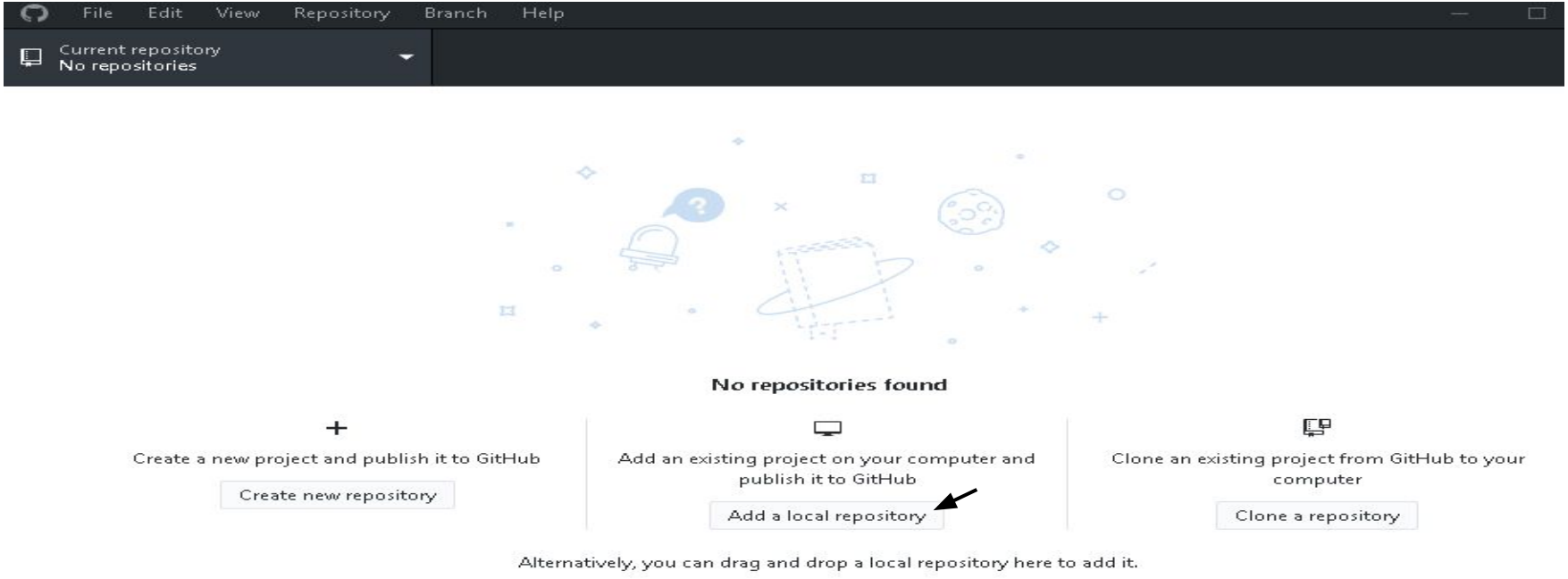






Choose Repository

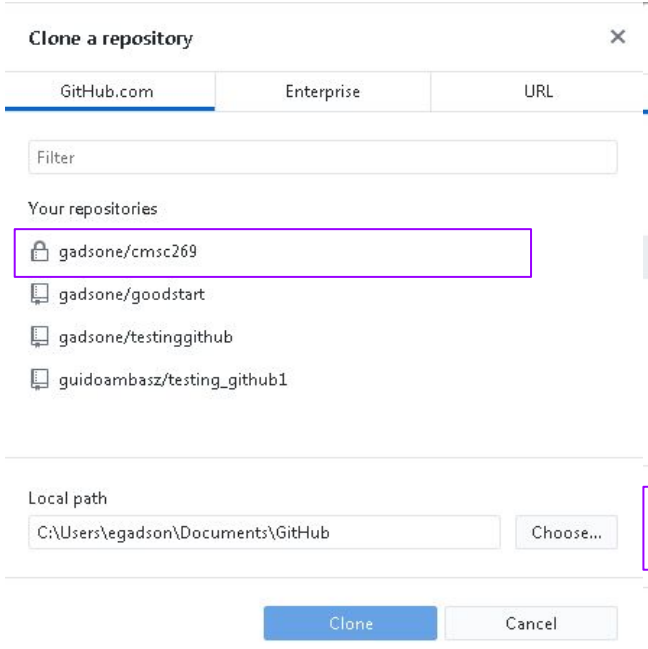
Select “Add a Local Repository” to clone a repository from GitHub.com. You can also create a repository and clone one from your computer.

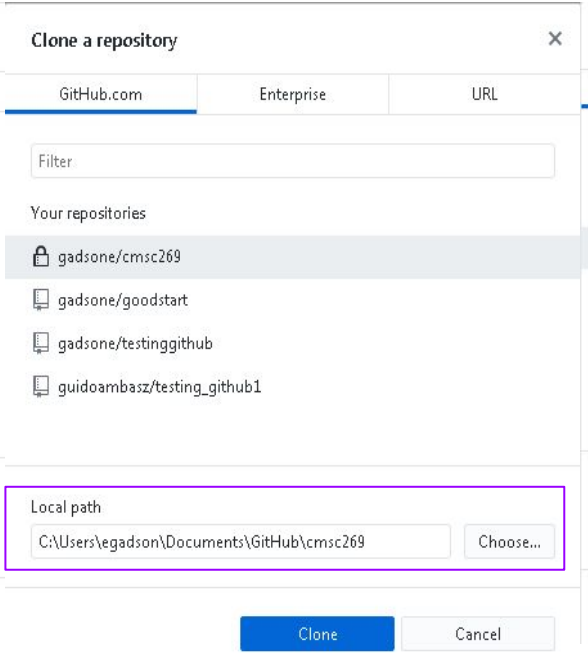


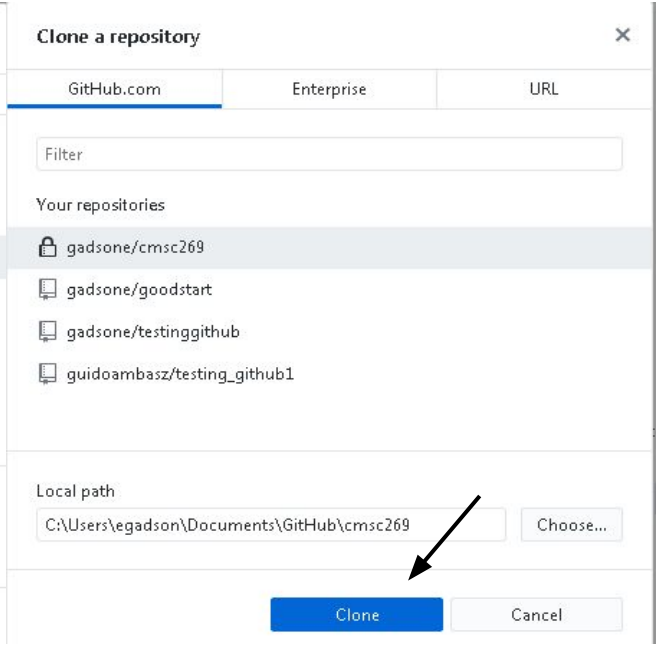
Select a repository from GitHub.com

Choose local path on your computer

Select “Clone”

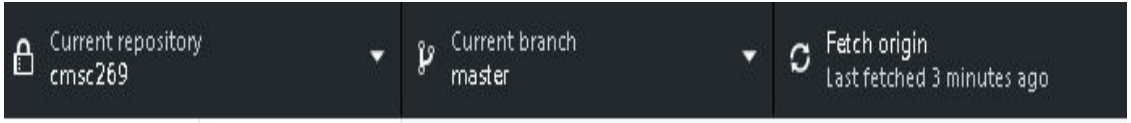






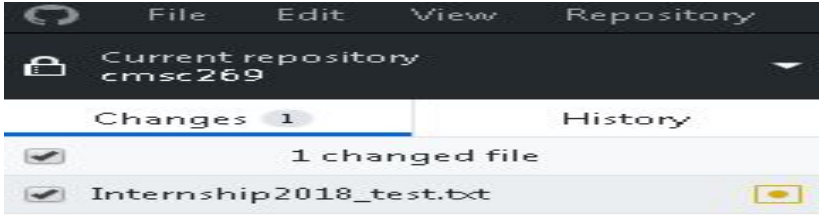
How to Use GitHub Desktop

- Click “fetch origin” to git pull (all information from GitHub.com

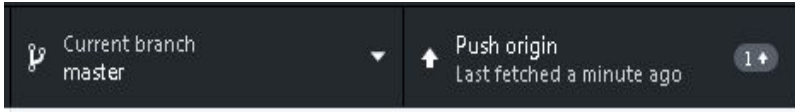


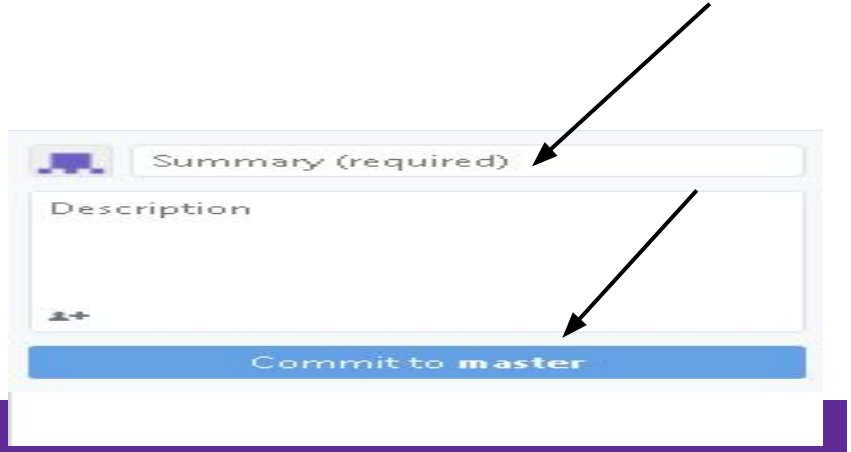
- Locate folder where your repository is located and add or change files. *(Remember local path on your computer)*

Files added/changes will show under “Changes” tab.



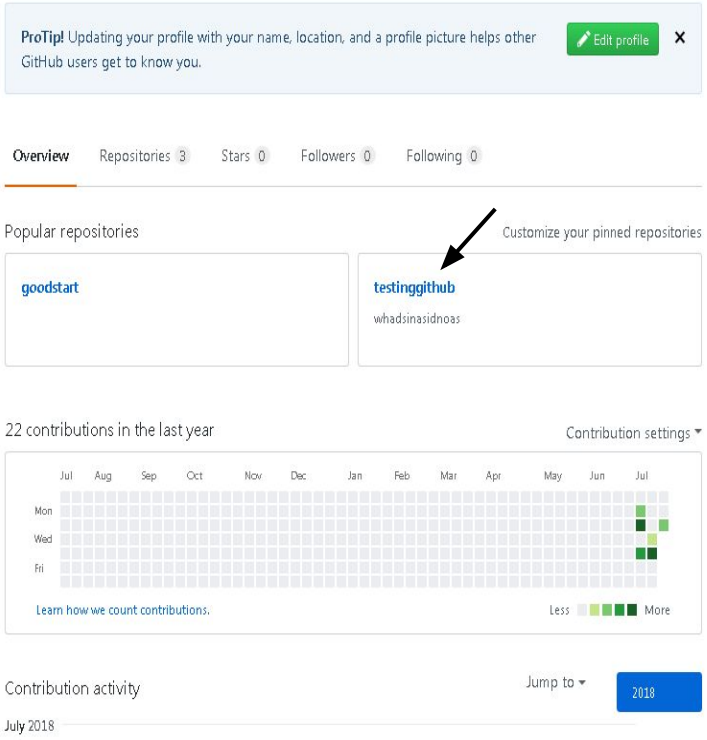
- Add a summary of what was changed (a message)
- You can add a description but it is not necessary
- Click “Commit to [branch]”
- Select “Push Origin” to push changes to GitHub.com

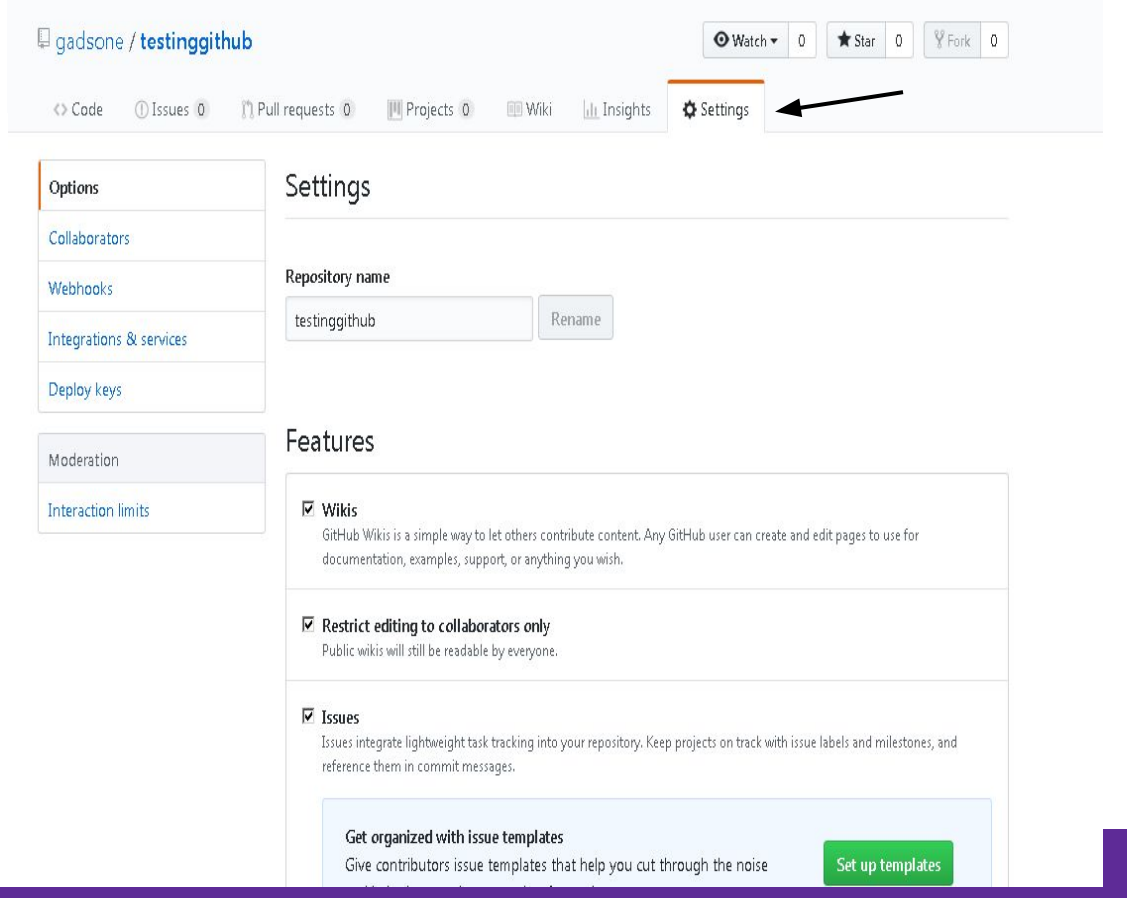




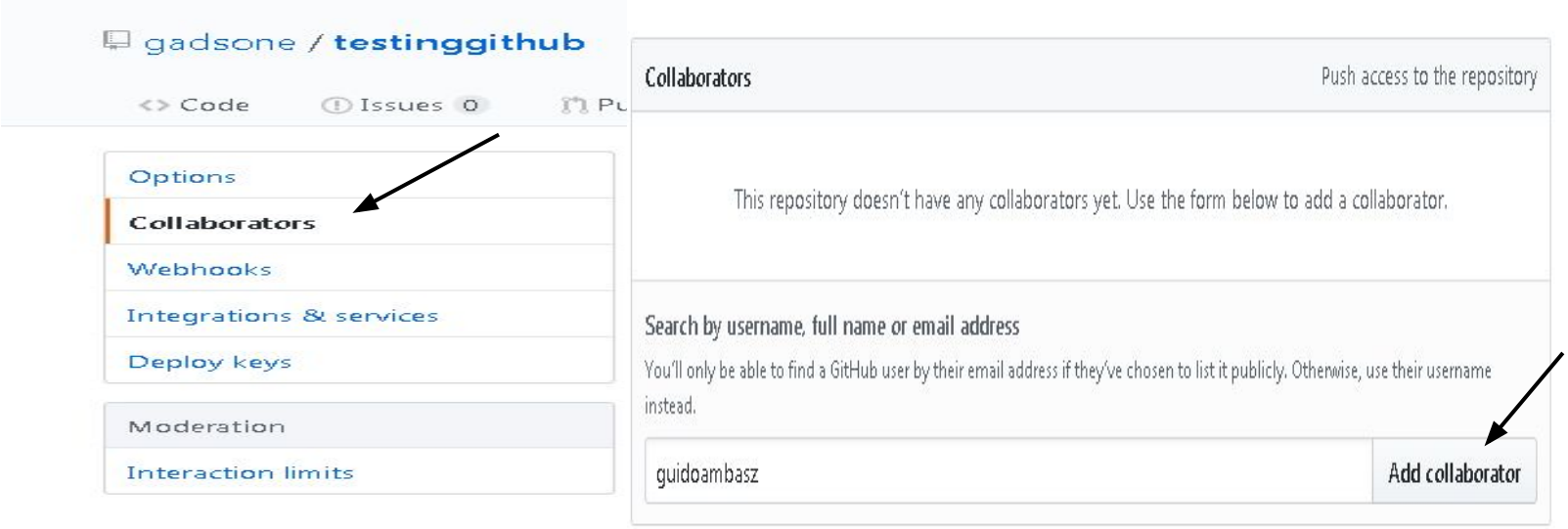
How to Add Collaborators in GitHub.com

Select a repository and choose the “settings” tab.

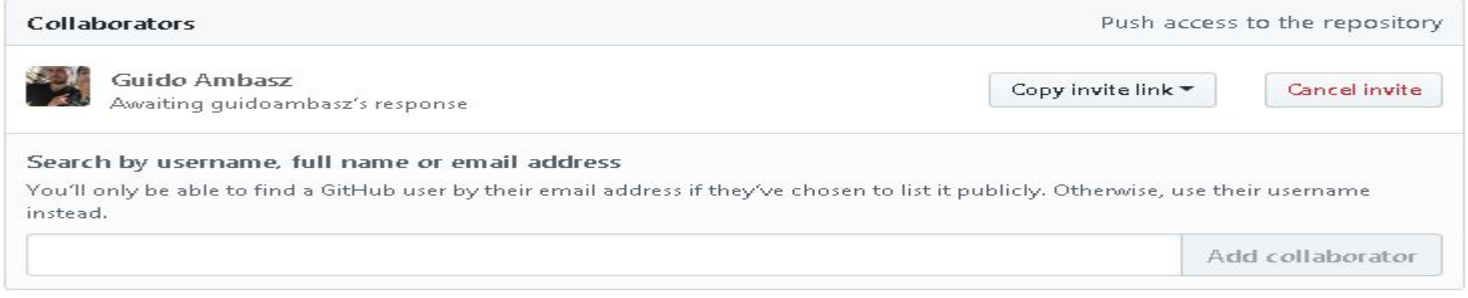




Choose “Collaborators” tab and type in username of co-author(s). Select add collaborator

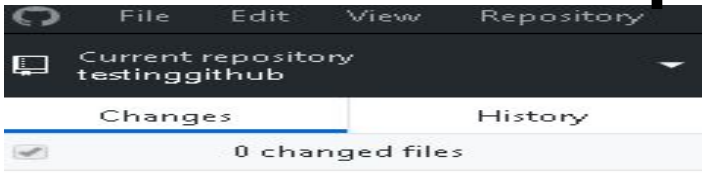


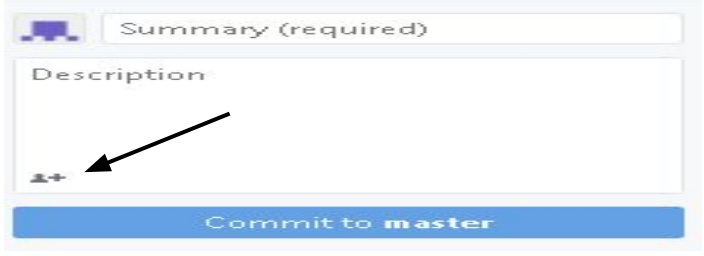
Co-author(s) should check their email to confirm invite.



How to Add Collaborators in GitHub Desktop

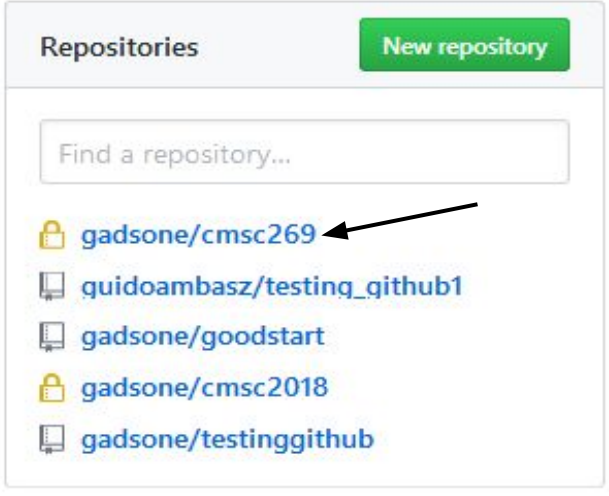
You can also add co-author(s) by selecting the “add co-authors” icon located in the description box each time you commit a change in a branch.

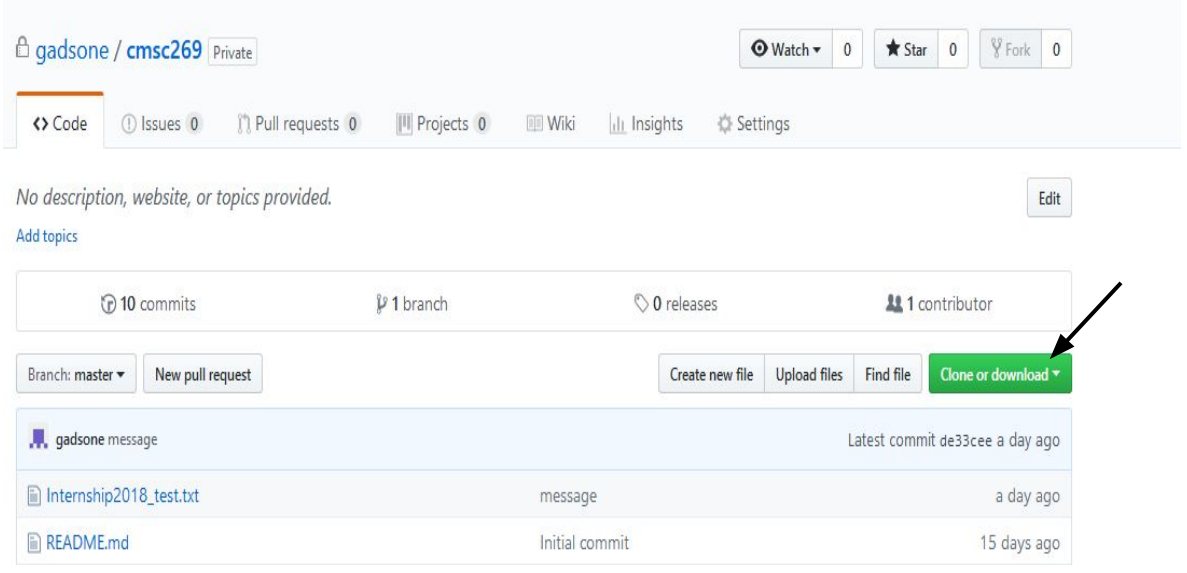




Download the Repository to Your Computer

After you click on the link (Clone or Download) you should be able to see this page. Click on ‘Clone or download’ and copy the link:



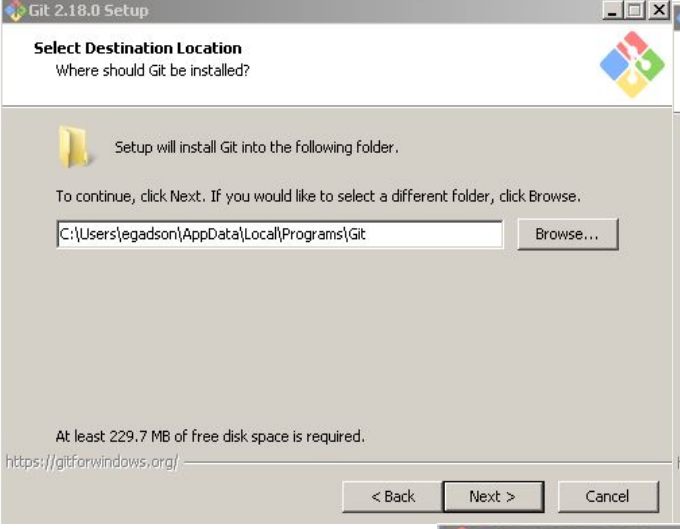


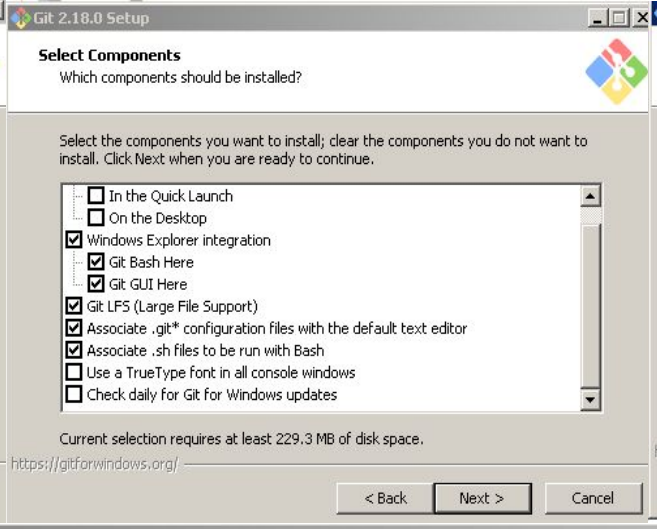
Download GitBash

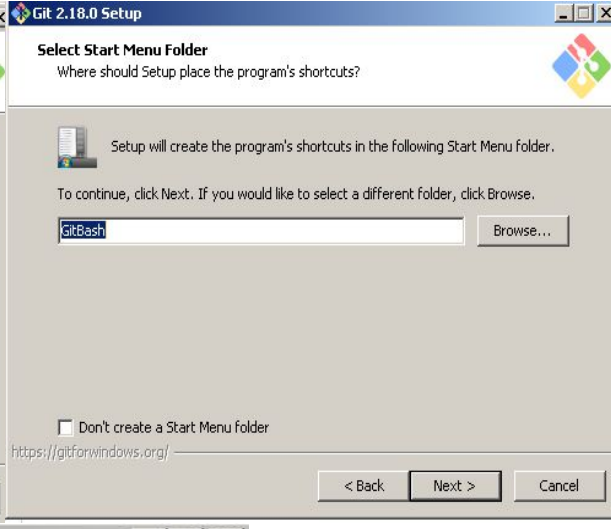
Download GitBash, a repository control system for your computer.

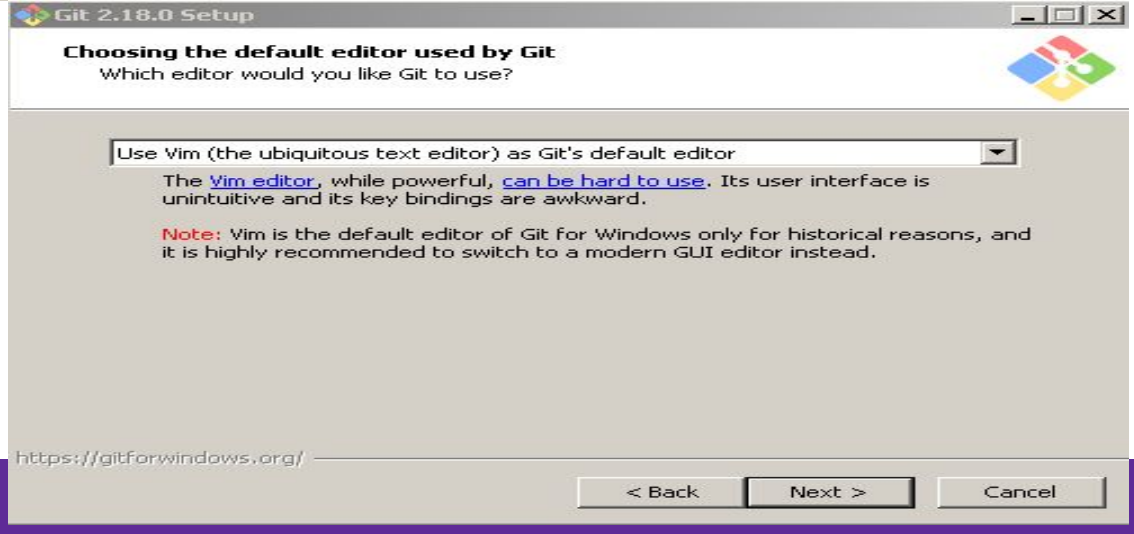
https://git-scm.com/download/win

Then complete the installation with the following steps:

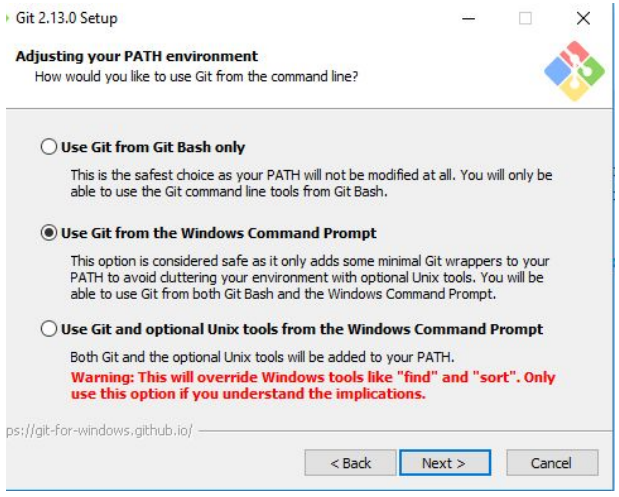

Git 2.18.0 Setup
Select Destination Location
Where should Git be installed?
Setup will install Git into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Users\egadson\AppData\Local\Programs\Git
Browse...
At least 229.7 MB of free disk space is required.
https://gitforwindows.org/
< Back
Next >
Cancel

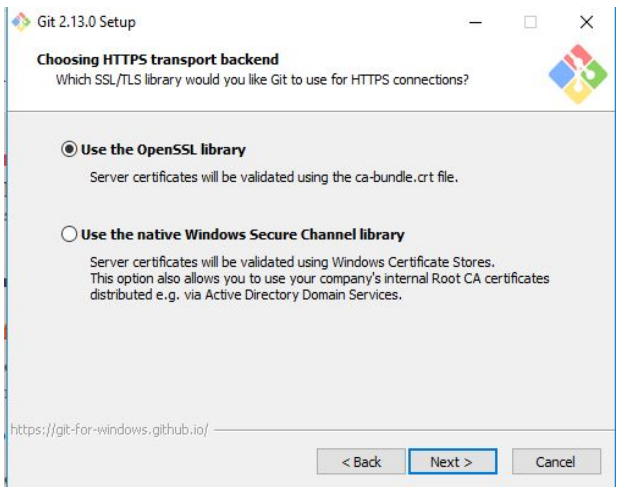

Git 2.18.0 Setup
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
In the Quick Launch
On the Desktop
Windows Explorer integration
Git Bash Here
Git GUI Here
Git LFS (Large File Support)
Associate .git* configuration files with the default text editor
Associate .sh files to be run with Bash
Use a TrueType font in all console windows
Check daily for Git for Windows updates
Current selection requires at least 229.3 MB of disk space.
https://gitforwindows.org/
< Back
Next >
Cancel

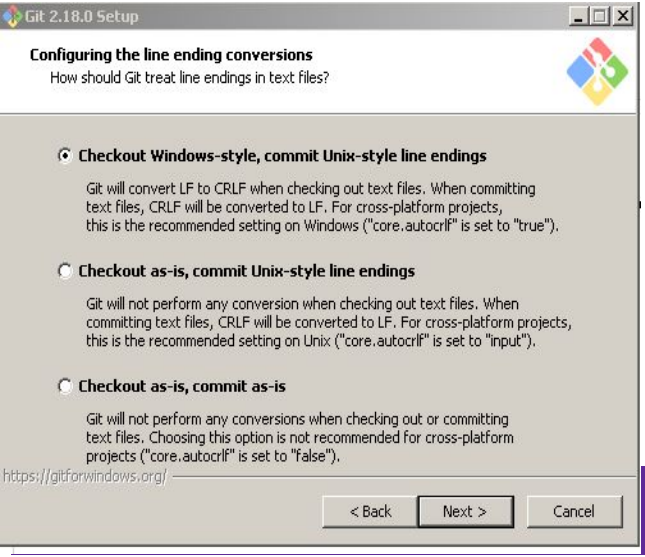

Git 2.18.0 Setup
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
GitBash
Browse...
Don't create a Start Menu folder
https://gitforwindows.org/
< Back
Next >
Cancel

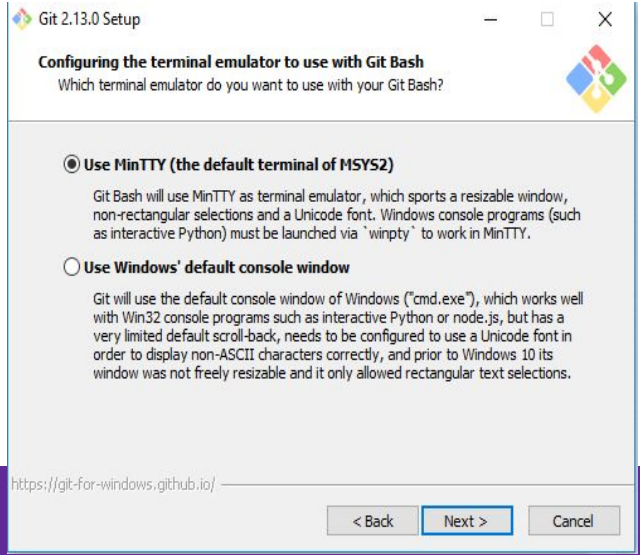

Git 2.18.0 Setup
Choosing the default editor used by Git
Which editor would you like Git to use?
Use Vim (the ubiquitous text editor) as Git's default editor
The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.
Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.
https://gitforwindows.org/
< Back
Next >
Cancel

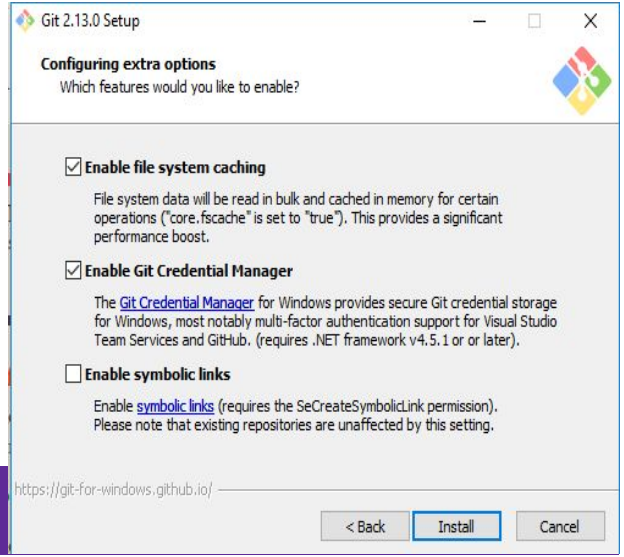

Git 2.13.0 Setup
Information
Please read the following important information before continuing.
When you are ready to continue with Setup, click Next.
GNU General Public License
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, 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
https://git-for-windows.github.io/
Next >
Cancel


Git 2.13.0 Setup
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
Use Git from the Windows Command Prompt
This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.
Use Git and optional Unix tools from the Windows Command Prompt
Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.
ps://git-for-windows.github.io/
< Back
Next >
Cancel


Git 2.13.0 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.
https://git-for-windows.github.io/
< Back
Next >
Cancel

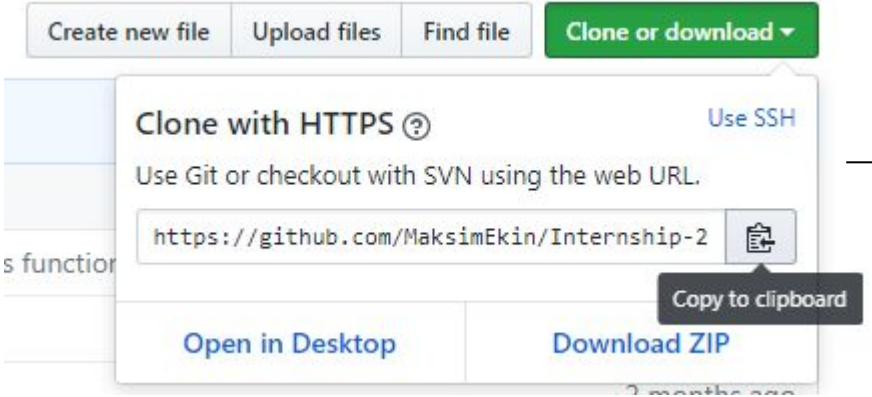

Git 2.18.0 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
https://gitforwindows.org/
< Back
Next >
Cancel

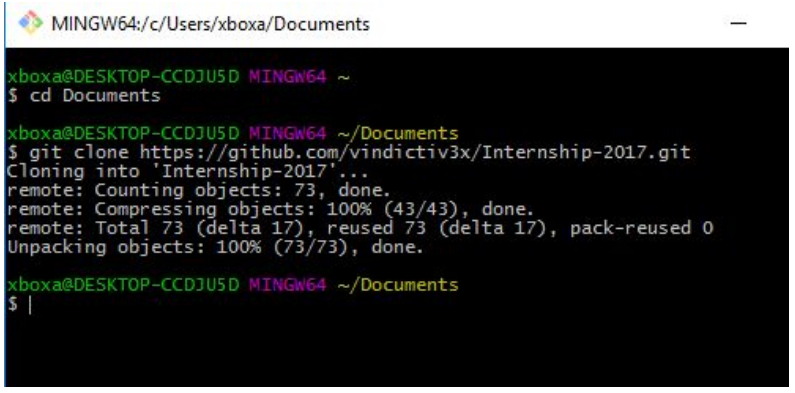

Git 2.13.0 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.
Use Windows' default console window
Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.
https://git-for-windows.github.io/
< Back
Next >
Cancel


Git 2.13.0 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 or later).
Enable symbolic links
Enable symbolic links (requires the SeCreateSymbolicLink permission).
Please note that existing repositories are unaffected by this setting.
https://git-for-windows.github.io/
< Back
Install
Cancel

GitBash

To download the contents, type in the command console “git clone (the link you copied from the website or from clone and download) and press enter” after you specify where you want to download your contents (follow screenshot). Go to the location you specified and make sure the folder is there.



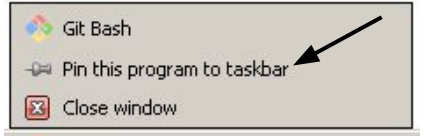


GitBash

After downloading the repository to the location you'd like to have as your workplace, use the start menu to open GitBash. You can pin to taskbar by right-clicking on the program.

1

2



How To Use GitBash

- First, change directory by entering in cd repository name.

```
egadson@RLCTC211C630272 MINGW64 ~
$ cd cmsc269
```

- Next, locate folder where your repository is located and add files.
- Then, git add changes you have made to your files.

```
egadson@RLCTC211C630272 MINGW64 ~/cmsc269 (master)
$ git add peopleallover.txt
```

Commit Changes and Push into GitHub

- Then git commit -m “notes” to add message and commit the changes to the repository.

```
egadson@RLCTC211C630272 MINGW64 ~/cmsc269 (master)
$ git commit -m "made changes"
[master 74a4776] made changes
```

- Lastly, ‘git push” to push your changes to the repository.

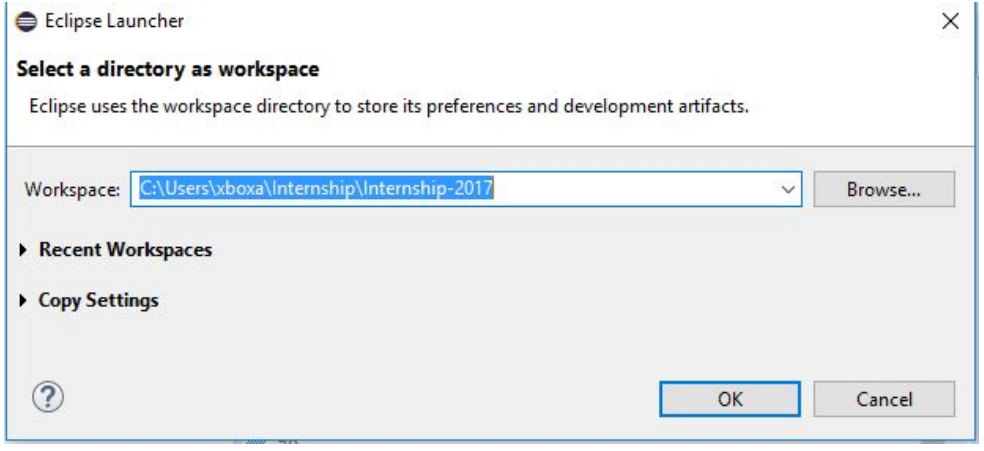
```
egadson@RLCTC211C630272 MINGW64 ~/cmsc269 (master)
$ git push
Enumerating objects: 6, done.
```

Using Git Pull

- ‘Git pull’ will pull the data on the repository. The changes will appear in your code. If there are changes in same line, eclipse will show it and let you choose the one you’d like to use.

GitBash - Eclipse Workspace

Choose your Eclipse workspace (same place where you cloned your repository to your computer)



GitBash - Eclipse Workspace

Create all the projects with same names in Eclipse manually. Codes should appear after you manually create the projects.

References & Quick Links

Educational Resources for Students: https://education.github.com/pack , https://education.github.com/

GitHub: https://github.com/

GitBash: https://git-scm.com/download/win

GitHub Desktop: https://desktop.github.com/

GitHub Glossary: https://help.github.com/articles/github-glossary/

GitHub Youtube Channel: https://www.youtube.com/user/GitHubGuides

The End