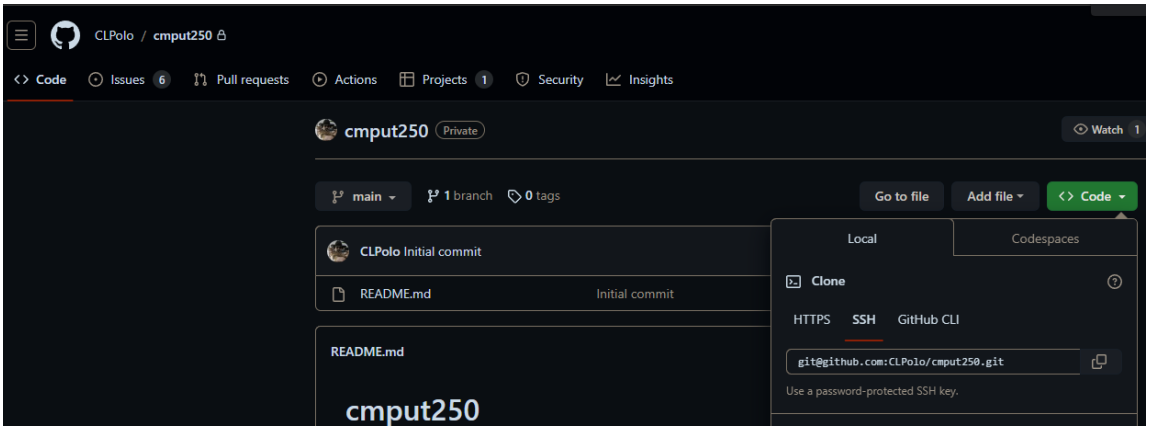
Part 1: Installing Git (if you don't already have it)

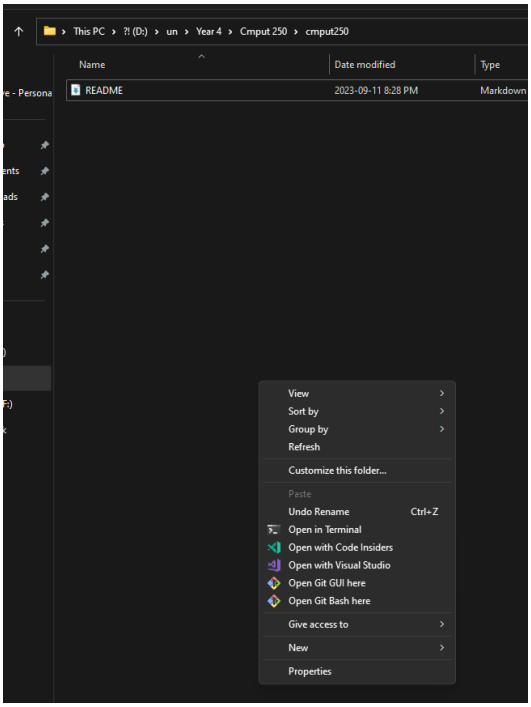
1. https://git-scm.com/ download the installer from here.
2. Run the installer.
    a. Ensure git bash is installed (git bash shows branch names beside the directory path)
3. That's it you are done!

Part 2: Accessing the git repository

1. Go to the repository: https://github.com/CLPolo/cmput250
2. If you do not already have a github ssh key set up, you will need to set up an ssh key and register it on your device, see the following github documentation: https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent
3. Copy the ssh link from the code section:



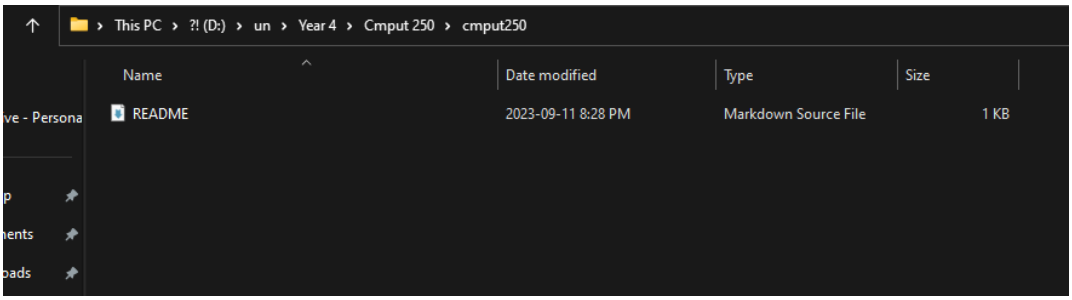
4. Open git bash (right click > show more options first) where you want the repository to be located on your local system:



5. Type `git clone <ssh-link>` replacing <ssh-link> with the link previously copied

```
Gurtaj K@Desktop310 MINGW64 /d/un/Year 4/Cmput 250
$ git clone git@github.com:CLPolo/cmput250.git
```

6. The files will now be there, you are done!



Part 2: Adding to the git repository

1. Traverse into the repository using `CD: cmput250` from the previous folder

```
Gurtaj K@Desktop310 MINGW64 /d/un/Year 4/Cmput 250
$ cd cmput250

Gurtaj K@Desktop310 MINGW64 /d/un/Year 4/Cmput 250/cmput250 (main)
$
```

2. Create a branch for your changes using `git branch <branch-name>` (detailed names help) and switch to it `git switch <branch-name>
    a. Alternatively use `git brsw <branch-name>` to create and switch to it (assuming config is set up, see last page for git configs)
    b. **NOTE:** branches are created from the **CURRENT** branch, switch to the main branch before you create new branches, it's also common practice to pull all changes before creating the branch to prevent merge conflicts in the future

```
Gurtaj K@Desktop310 MINGW64 /d/un/Year 4/Cmput 250/cmput250 (main)
$ git branch adding-git-instructions-to-repo

Gurtaj K@Desktop310 MINGW64 /d/un/Year 4/Cmput 250/cmput250 (main)
$ git switch adding-git-instructions-to-repo
Switched to branch 'adding-git-instructions-to-repo'

Gurtaj K@Desktop310 MINGW64 /d/un/Year 4/Cmput 250/cmput250 (adding-git-instructions-to-repo)
$
```

3. Make any changes to the file system as needed and type `git add –all` to add your changes to the selection of tracked files, then type `git commit -m “<message>”` to create a commit

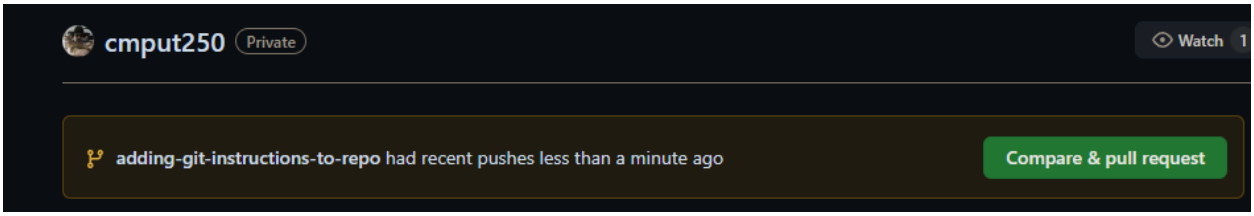
```
Gurtaj K@Desktop310 MINGW64 /d/un/Year 4/Cmput 250/cmput250 (adding-git-instructions-to-repo)
$ git add --all

Gurtaj K@Desktop310 MINGW64 /d/un/Year 4/Cmput 250/cmput250 (adding-git-instructions-to-repo)
$ git commit -m "added a word document with instructions on using git"
[adding-git-instructions-to-repo 03a984a] added a word document with instructions on using git
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Documentation/Git Setup Help.docx
 create mode 100644 Documentation/~$t Setup Help.docx
```

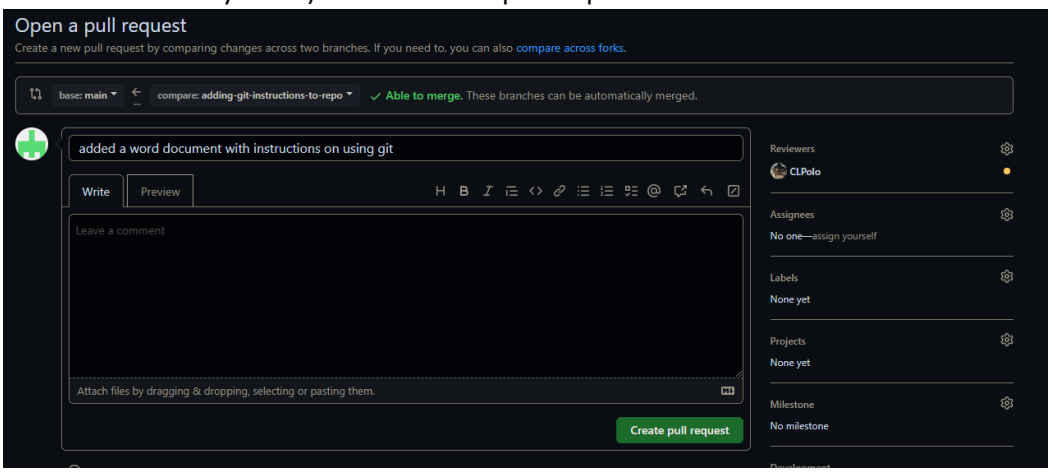
4. Push your changes using `git push –set-upstream origin <branch-name>` (or simply `git pushu` if config is set up [see last page])

```
Gurtaj K@Desktop310 MINGW64 /d/un/Year 4/Cmput 250/cmput250 (adding-git-instructions-to-repo)
$ git push --set-upstream origin adding-git-instructions-to-repo
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 6 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 141.49 KiB | 2.48 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'adding-git-instructions-to-repo' on GitHub by visiting:
remote:      https://github.com/CLPolo/cmput250/pull/new/adding-git-instructions-to-repo
remote:
To github.com:CLPolo/cmput250.git
 * [new branch]      adding-git-instructions-to-repo -> adding-git-instructions-to-repo
branch 'adding-git-instructions-to-repo' set up to track 'origin/adding-git-instructions-to-repo'.
```

5. Go onto the github repository and create a pull request for your branch to be merged with the main branch:

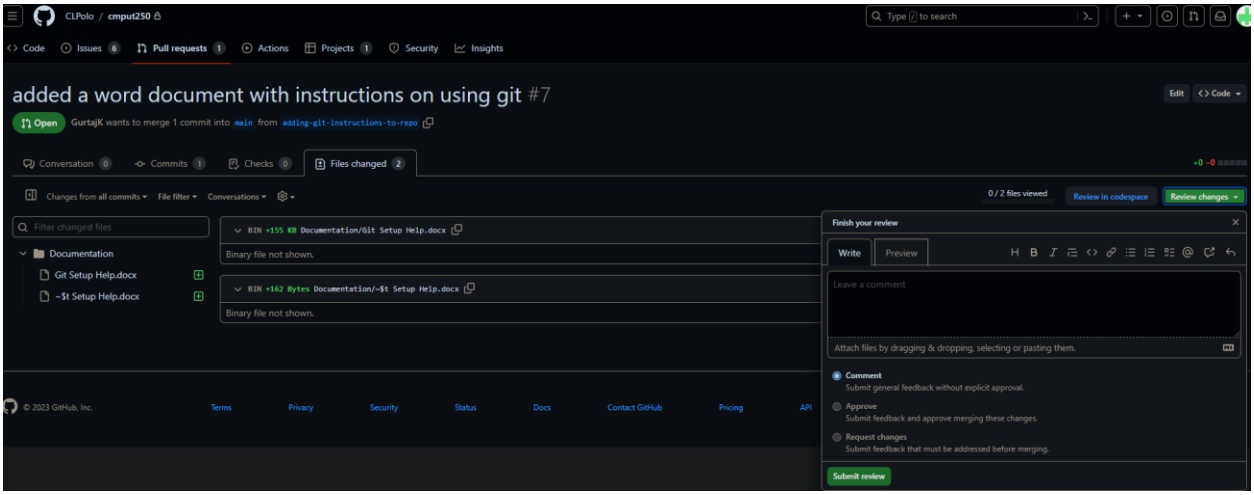


6. Add any reviewers, labels, link any issues (use #number in the comment section or the title and it will automatically link it) and create the pull request.



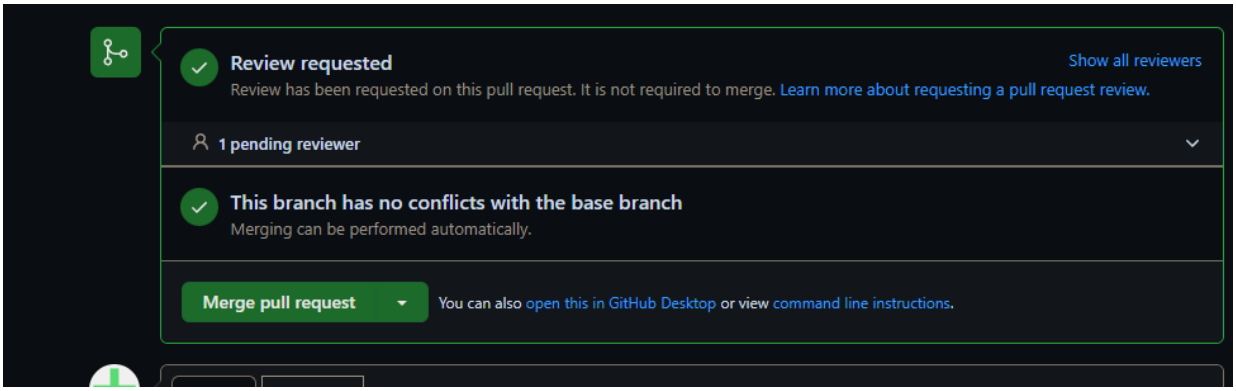
Reviewing PRs:

1. Add comments and either approve or request something to be changed:

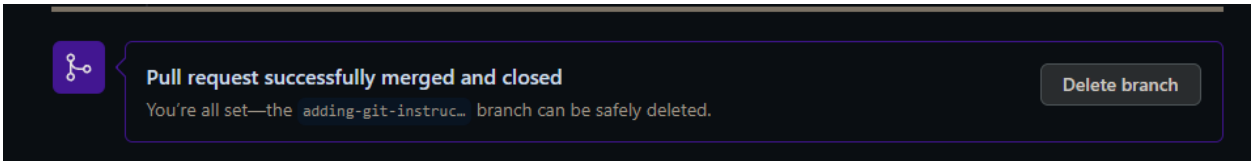


Merging PRs:

1. After changes from reviews have been made and the pr is approved, merge it by clicking merge:



2. Delete the excess branch after the pr gets merged to avoid clutter basically



Keeping your local files updated

1. Running a simple `git pull` will update the CURRENT branch with any changes in github
    a. If someone has made changes to the main branch that you cannot see, switch to the main branch and type `git pull`

(OPTIONAL): Set up git configs

1. Copy the contents in the .gitconfig file into your local .git/config file: