MAPÚA UNIVERSITY
SCHOOL OF ELECTRICAL, ELECTRONICS, AND COMPUTER ENGINEERING

Lab 1: Using Tools in Software Design

CPE106L (Software Design Laboratory)

Hannah Mae Antaran

Darrel Umali Virtusio

Kathleen Joy Tupas

Group: 01

Section: E01

PreLab

Readings

Conda Cheat Sheet. (2017, August 8). Retrieved June 12, 2020 from https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/condacheatsheet.pdf?fbclid=IwAR12QNIAsjvSvyk5ZaJNWXuhkW9vjpKEer8s6O1CChI4GMF_KvxYcyLWk54

GitHub Guides. (2016, April 7). Retrieved June 12, 2020, from https://guides.github.com/activities/hello-world/

Insights and Reflection

Github is a web-based code hosting platform used for version control and collaboration where users can work together on their projects. It is a platform where users can dwell more into using git commands as a basis for creating branches, making commits, and merging pull requests. GitHub, as mentioned in this reference, has a built-on Git where users do not need to install git itself. Some insights acquired upon reading:

- **Repositories** - used to organize a project. It contains folders that can store files, images, videos, spreadsheets, data sets, and anything that is needed for a project.
- **Branch** - used for bug fixes and feature work. Has similar purpose with repositories but branches are made to work on different versions of a repository which is called branching. In simpler words, branches are multiple versions of a repository to let users work on multiple aspects of their project. By default, GitHub has a branch named master branch which is considered the "final" branch made by the user for their project. This branch is anything but special as it is the same with other branches, however, this branch is called master as it contains the last commits made by the users.
- **Make and commit changes** - committing in GitHub means to save changes. Every commit as stated in this reference has a commit message wherein users who commit may add a note or explanation in line to their made changes.
- **Open a pull request** - when a user opens a pull request, they have already made a commit and are now suggesting those commits for other people to review or pull in their contribution to those other people's branch.
- **Merging pull request** - by means, merging pull requests is bringing your commits together and putting it into the master branch. In this experiment, merging pull requests was the last step for the free training the group has enrolled in.

In reflection to this reference and the experiment itself, students were entirely introduced to GitHub and how it works. This URL was provided upon the first activity made for this experiment, which was to explore GitHub. In prior reference to this URL, the instructor directed us to enter a tutorial where users who try out the training meets the objective of familiarization to GitHub whereas enrollees are tasked to assign themselves to an issue, setting up their GitHub in reference to a specified issue, creating branches, committing files, opening a pull request, making a proper respond to a review, and merging pull requests. All of these are done through git commands made in anaconda prompt.

InLab

- ## Objectives

1. Demonstrate some of the few features of Github.
2. Create, activate and deactivate environments in Anaconda.
3. Use git commands to make commits, pull a request, and merge files.

- ## Tools Used

1. Anaconda
2. Git Terminal

- ## Procedure

FAMILIARIZATION WITH ANACONDA ENVIRONMENT

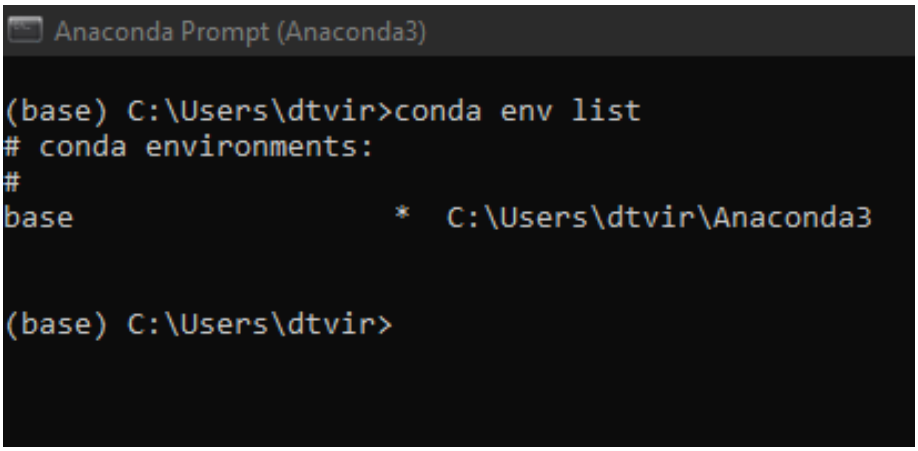


Figure 1.1 *Listing the current environment.*

We checked first if there is a created environment in Anaconda. There is no created environment so we will create one on the next step.

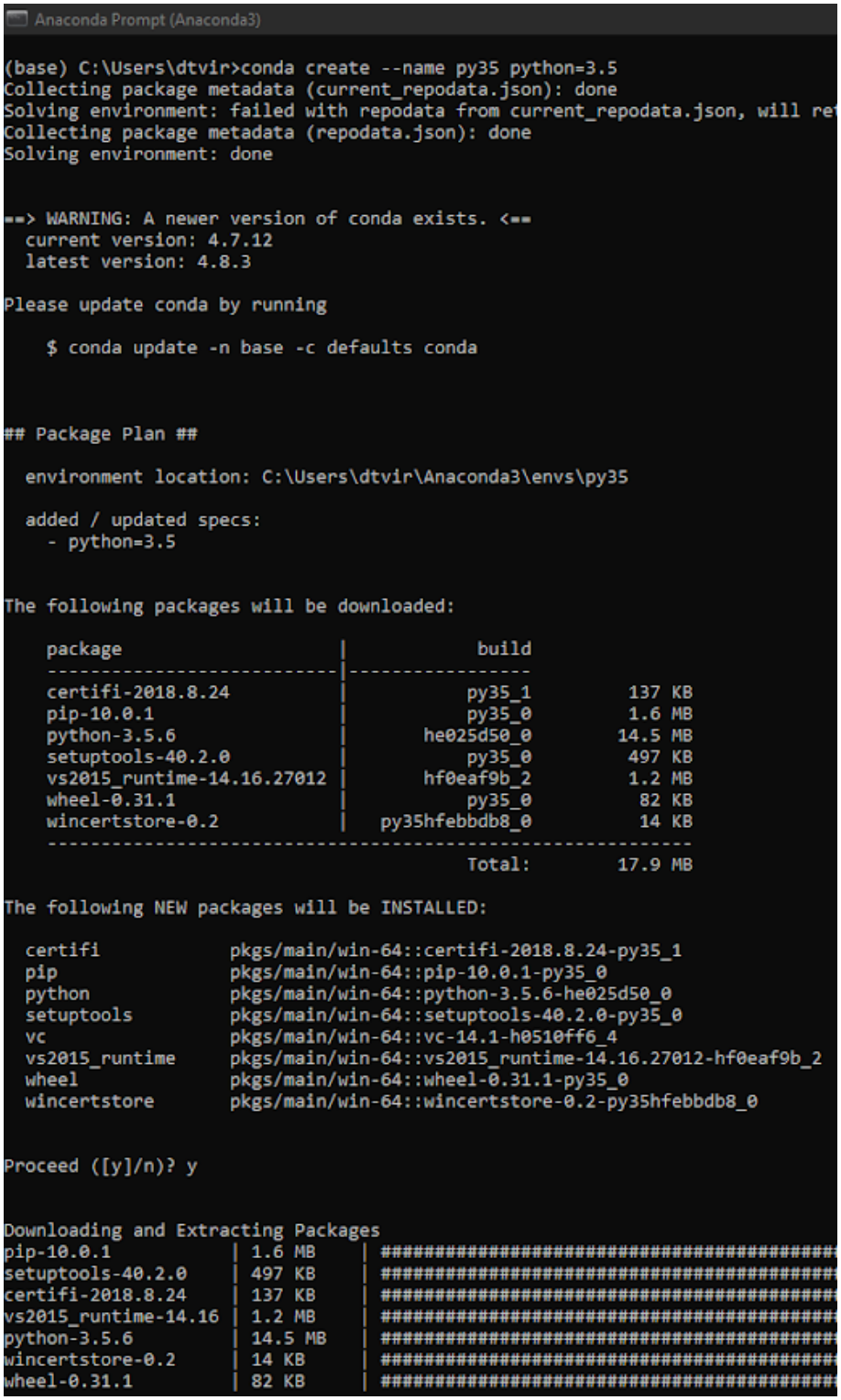


Figure 1.2 *Creating the new environment, py35.*

We successfully created the environment named py35 and installed a python with version 3.5 on the environment.

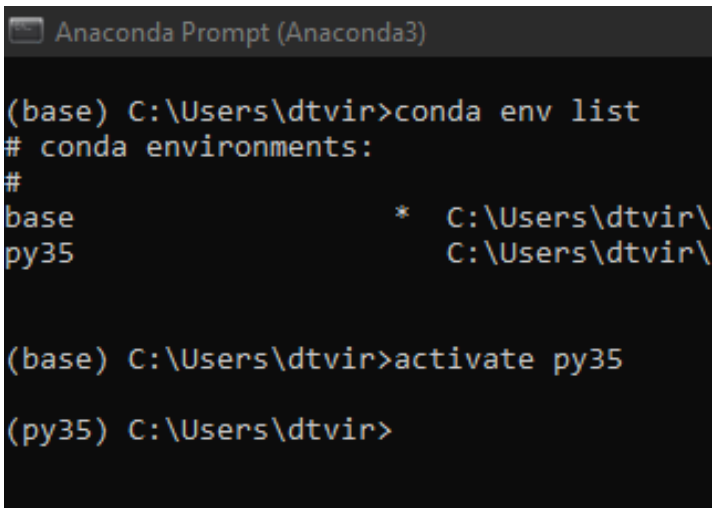
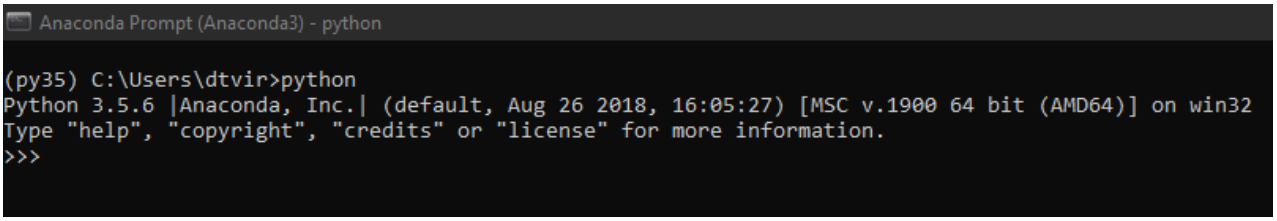
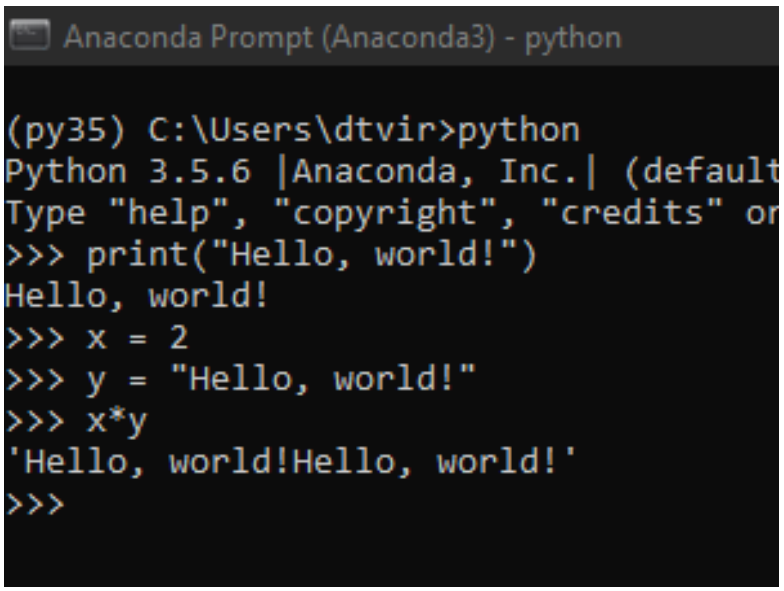


Figure 1.3 *Activating the py35 environment.*

We activate py35 environment using the "activate" command.





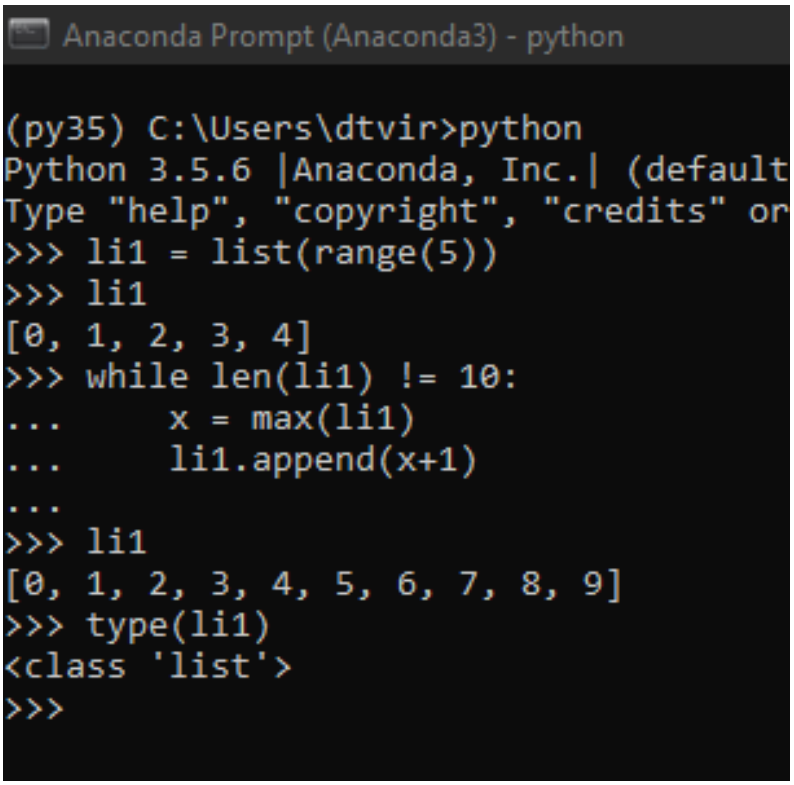


Figure 1.4 *Running python on py35 environment.*

We ran python on the environment and test it with a few codes like printing "Hello, world!", assigning a value to variables, creating a list, and using some functions like in list.

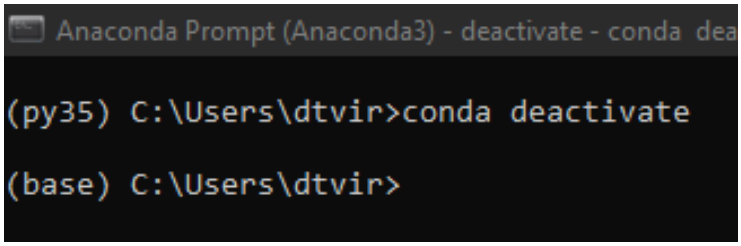


Figure 1.5 *Deactivating the environment.*

We deactivated the environment after using it.

FAMILIARIZATION WITH GITHUB ENVIRONMENT

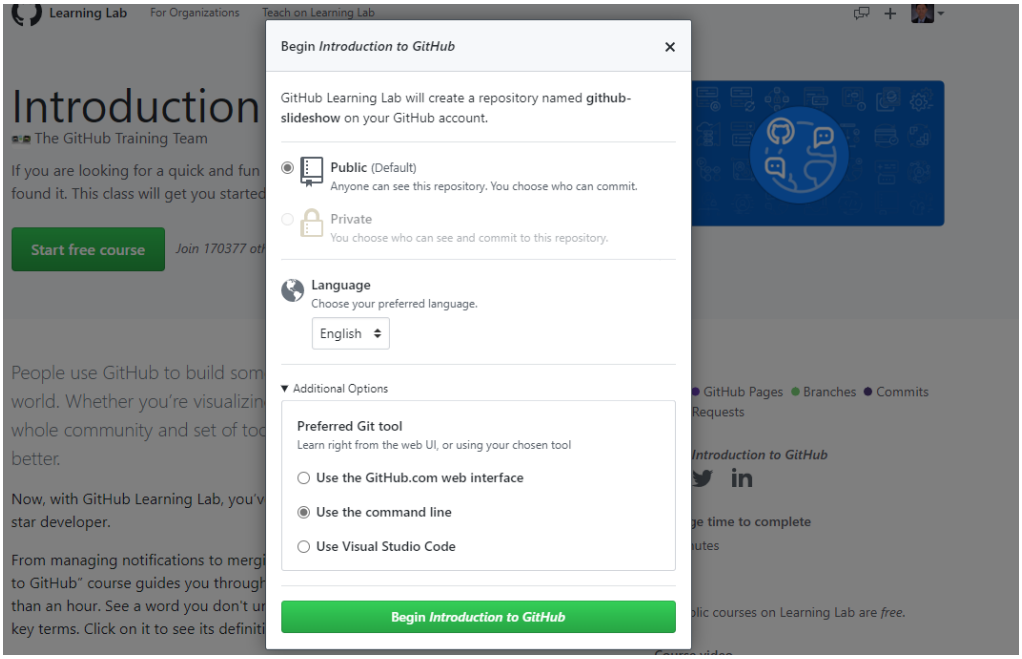


Figure 2.1 *Using GitHub on the command line.*

The tool that we will use on learning the GitHub is the command line. The interface shown is the GitHub learning lab.

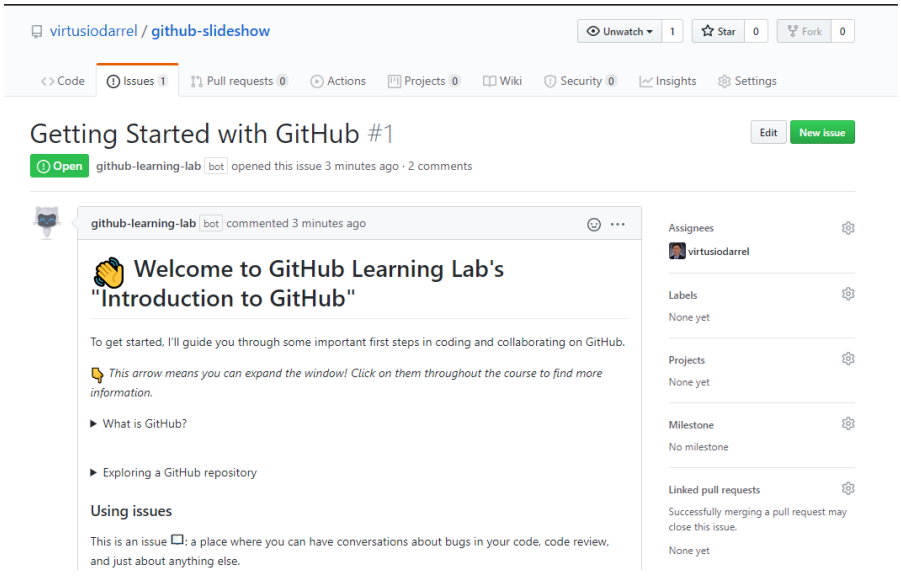


Figure 2.2 *Getting started with GitHub.*

The first task is to assign yourself and we have successfully done it.

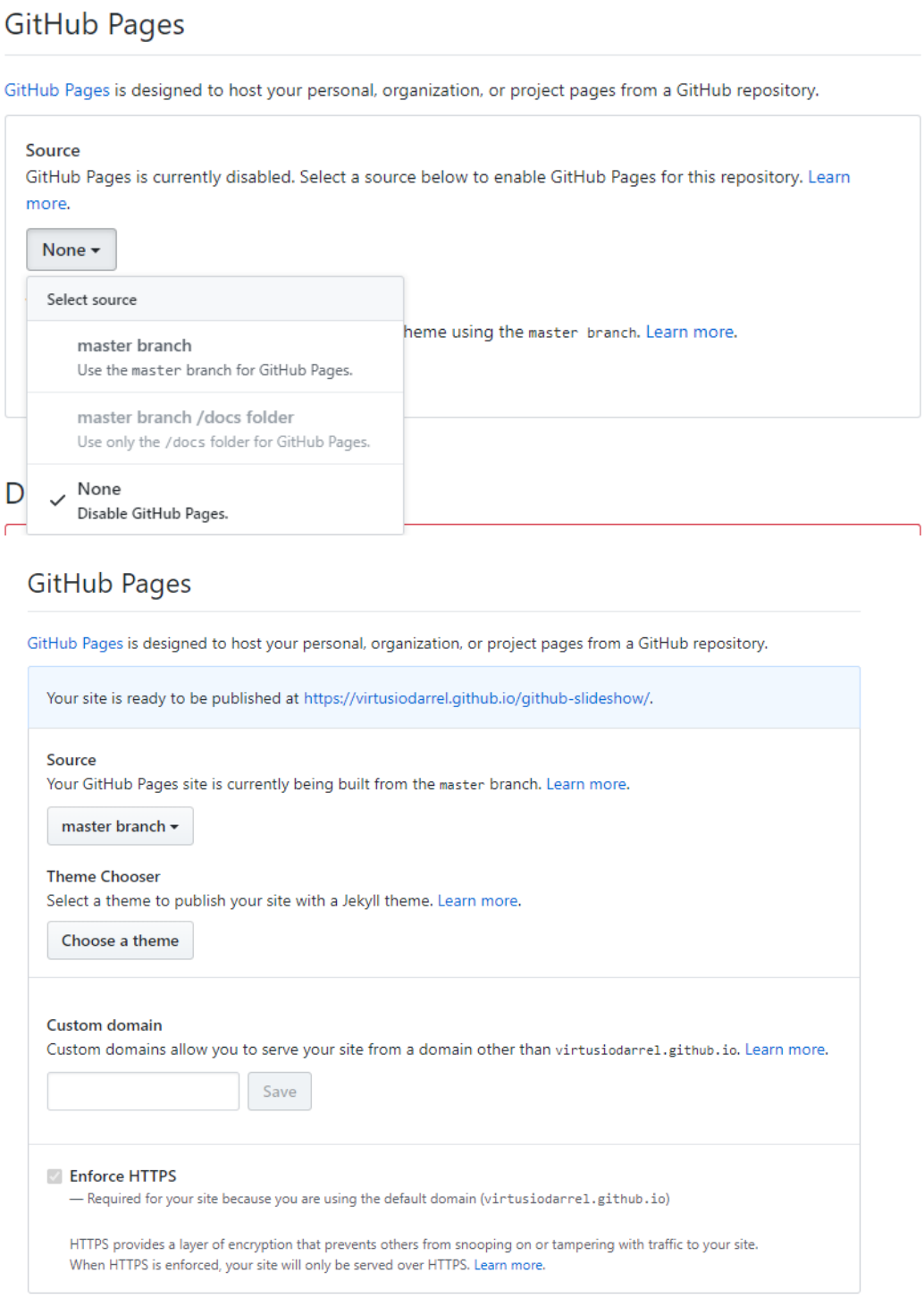


Figure 2.3 *Turning on GitHub pages.*

We turned on GitHub pages and it is currently being built from the master branch.

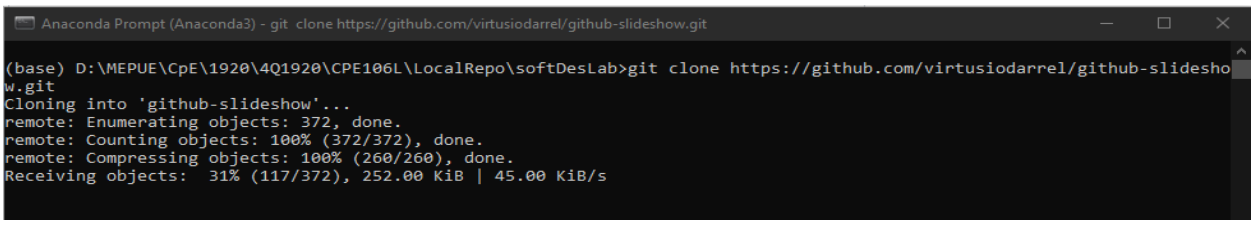


Figure 2.4 *Cloning github-slideshow repository.*

In this step, we clone the repository github-slideshow using the git clone command as we progress on the tutorial.

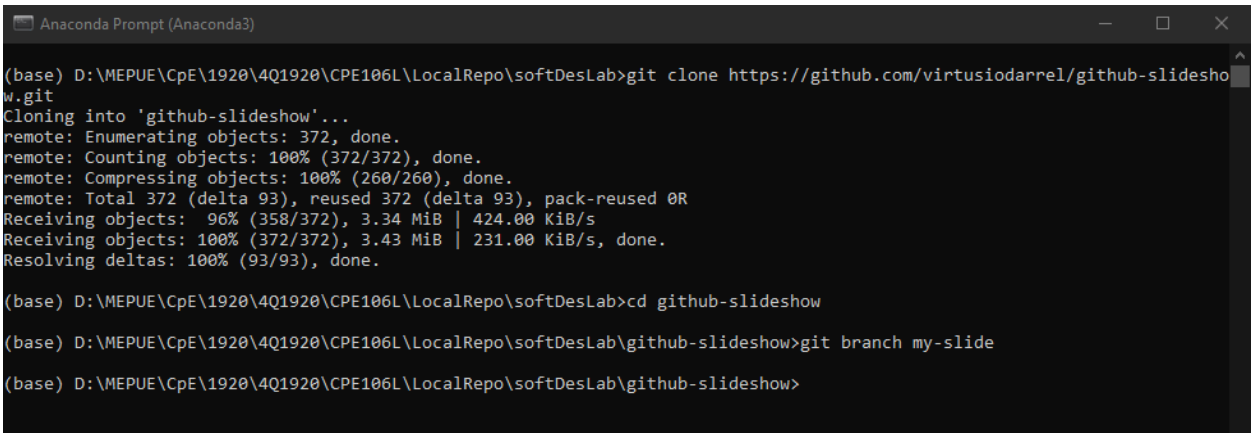


Figure 2.5 *Creating a github branch, my-slide.*

Using the git branch command, we created a new branch called my-slide where we will make all the changes necessary on the tutorial.

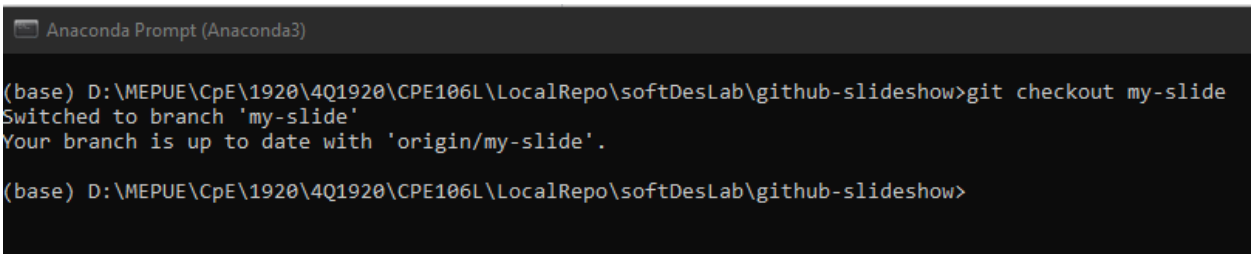
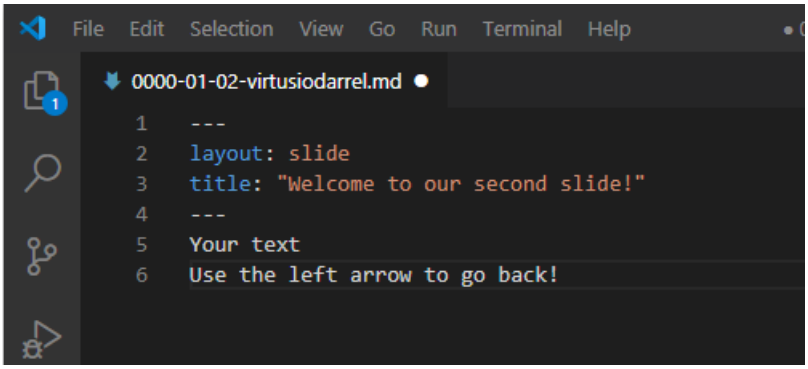


Figure 2.6 *Switching branch from master to my-slide.*

In this part, we changed from master branch to the created my-slide branch using the command git checkout.




```
(base) D:\MEPUE\CpE\1920\4Q1920\CPE106L\LocalRepo\softDesLab\github-slideshow>git add _posts/0000-01-02-virtusiodarrel.m
d
```

Figure 2.7 *Creating and adding a file to branch my-slide.*

After switching branches, we created the new .MD file and add it to the git repository using the git add command.

```
(base) D:\MEPUE\CpE\1920\4Q1920\CPE106L\LocalRepo\softDesLab\github-slideshow>git commit -m "add 0000-01-02-virtusiodarrel.md"
[my-slide c599a9b] add 0000-01-02-virtusiodarrel.md
 1 file changed, 6 insertions(+)
 create mode 100644 _posts/0000-01-02-virtusiodarrel.md

(base) D:\MEPUE\CpE\1920\4Q1920\CPE106L\LocalRepo\softDesLab\github-slideshow>
```

Figure 2.8 *Committing the file.*

We commit the added file using the git commit command and add a message about what is being changed on the repository. In this case, we added the .MD file.

```
(base) D:\MEPUE\CpE\1920\4Q1920\CPE106L\LocalRepo\softDesLab\github-slideshow>git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 471 bytes | 471.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/virtusiodarrel/github-slideshow.git
   5431165..c599a9b  my-slide -> my-slide
```

Figure 2.9 *Pushing the file to GitHub.*

Pushing files to GitHub means uploading the local repository content to a remote repository. We pushed the newly committed file using the git push command.

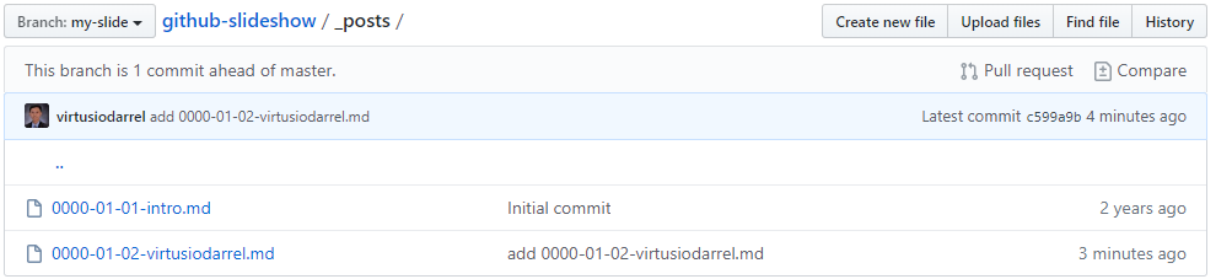


Figure 2.10 *New file added to Github.*

It shows that we have successfully pushed the .MD file because it was displayed in the my-slide branch on GitHub.

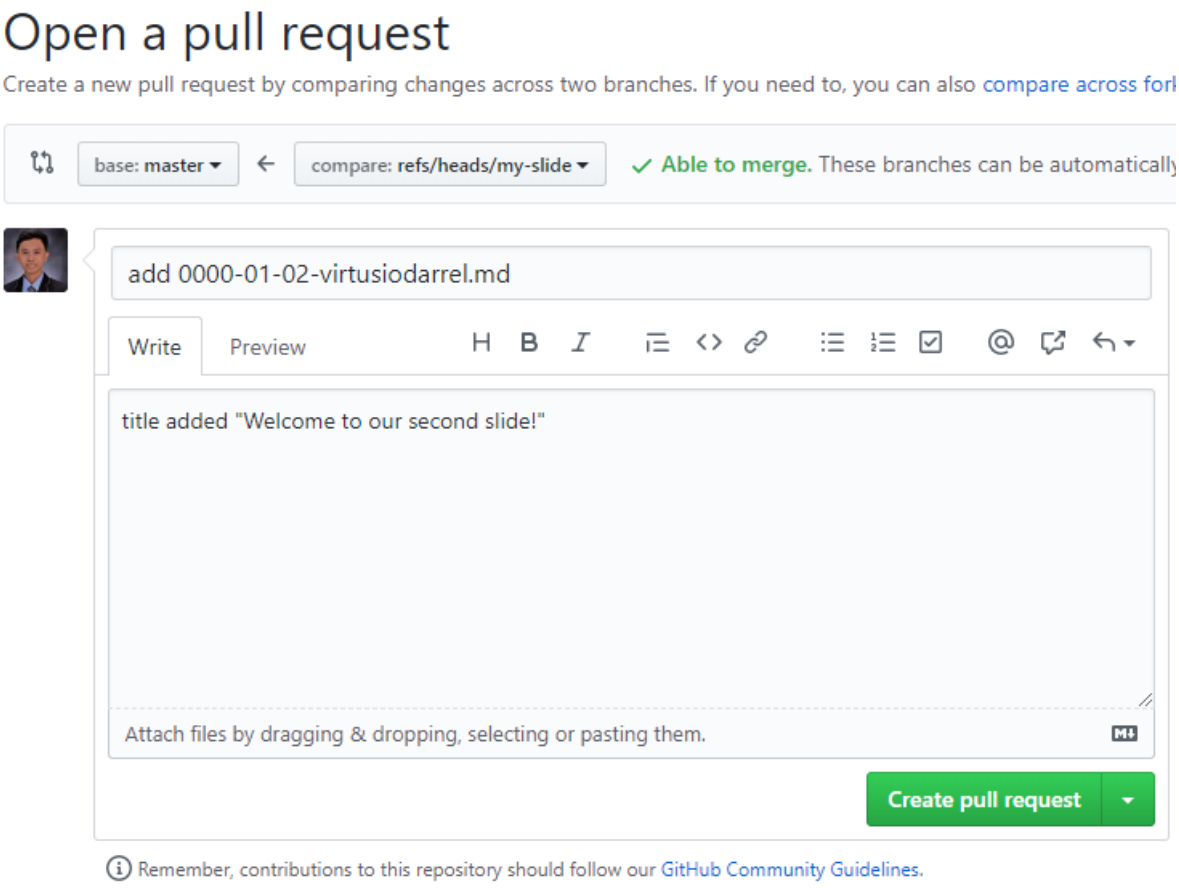


Figure 2.11 *Creating a pull request.*

A pull request means that it lets you tell other users on GitHub about the changes you have pushed to GitHub. This will open for review the set of changes, potential modifications, and possible future updates on the repository.

PostLab

FAMILIARIZATION WITH GITHUB ENVIRONMENT

```
(base) C:\LocalRepo\softDesLab\github-slideshow>git checkout my-slide
Already on 'my-slide'
Your branch is up to date with 'origin/my-slide'.

(base) C:\LocalRepo\softDesLab\github-slideshow>
```

Figure 3.1 *Switching branch to my-slide.*

In this part, we used git checkout to switch branch to my-slide.

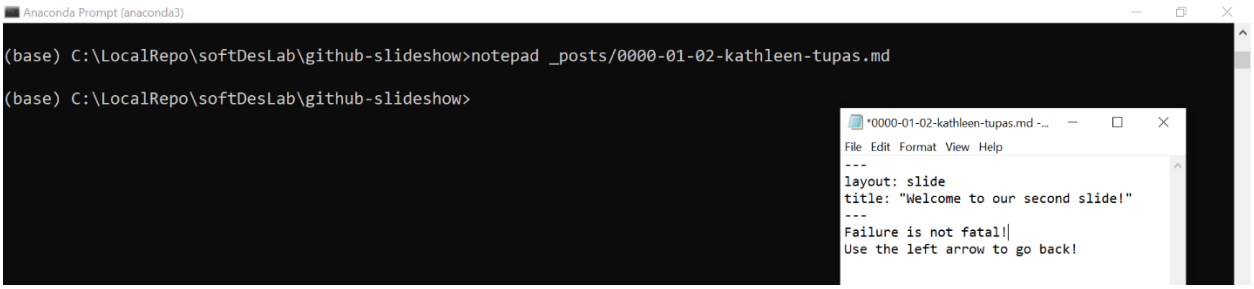


Figure 3.2 *Changing the content of the file.*

In this step, we are replacing line 5 of the file with something new.

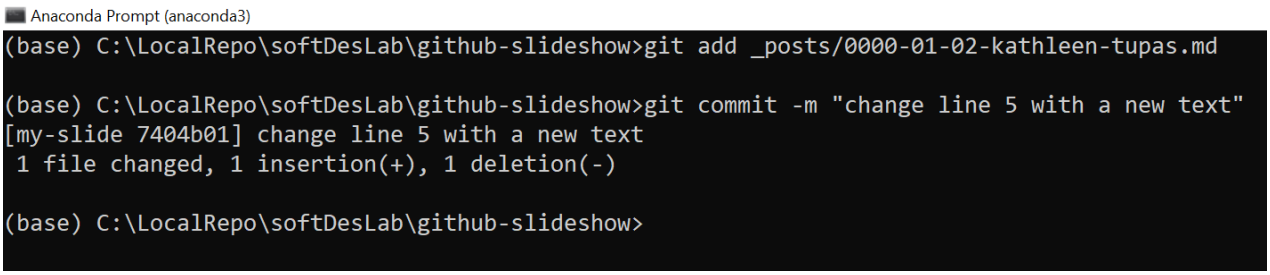


Figure 3.3 *Committing the file.*

We commit the added file using the git commit command and add a message about what is being changed on the repository.

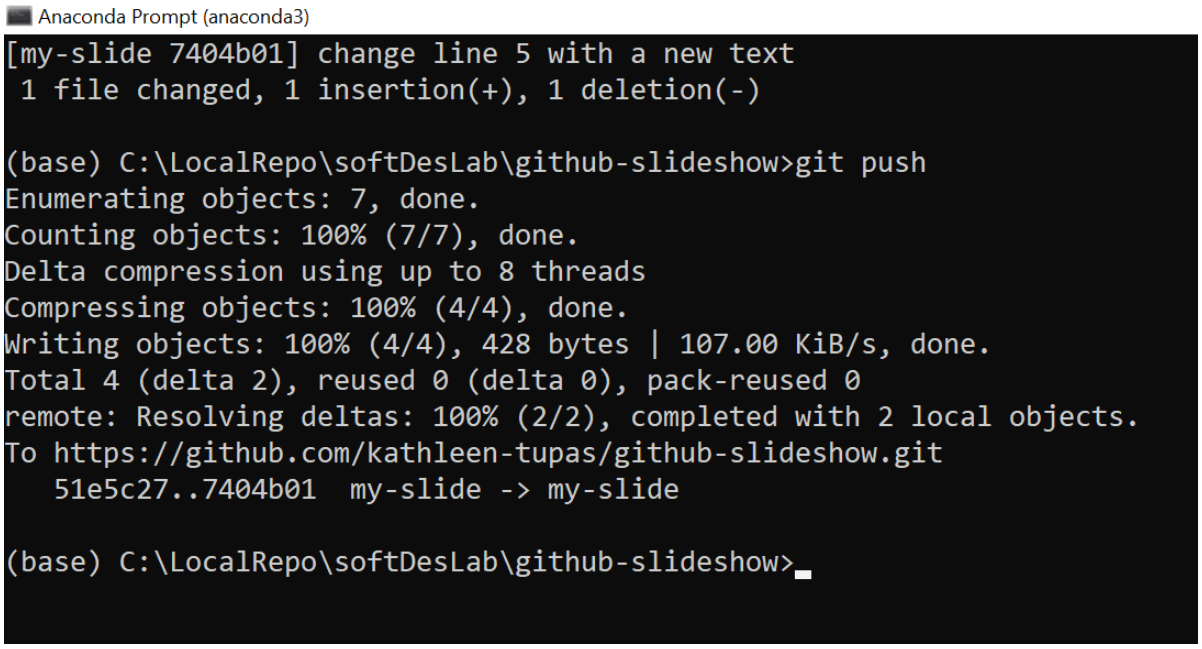


Figure 3.4 *Pushing the file to Github.*

We pushed the newly committed file using the git push command.

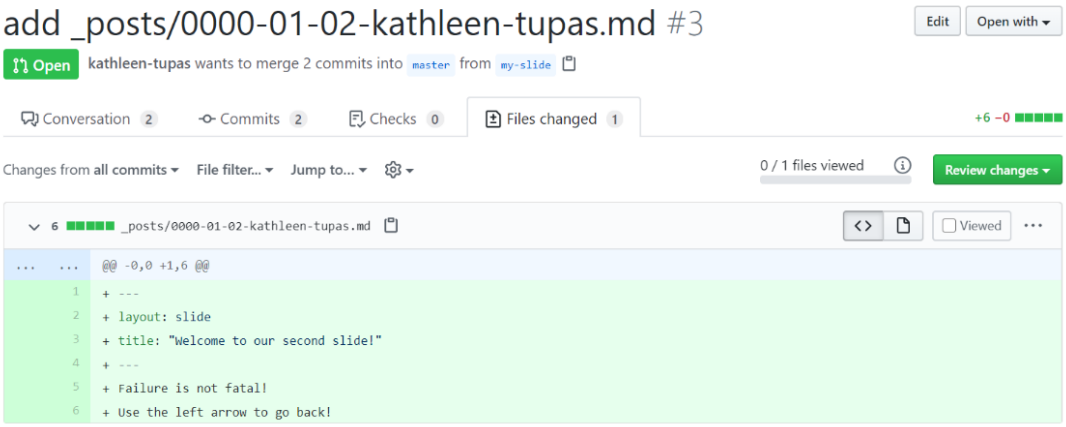


Figure 3.5 *The changes made were displayed in Github.*

It shows that we have successfully pushed the file because the changes made were displayed in the my-slide branch on GitHub.

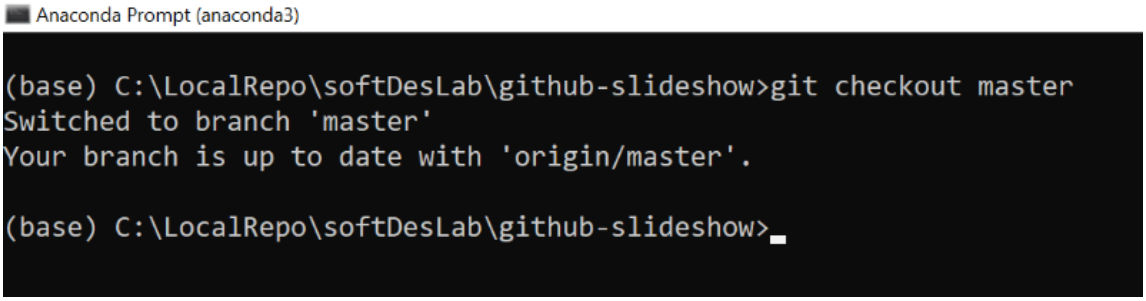


Figure 3.6 *Switching branch from my-slide to master.*

In this part, we switched from my-slide branch back to master branch to merge pull request.

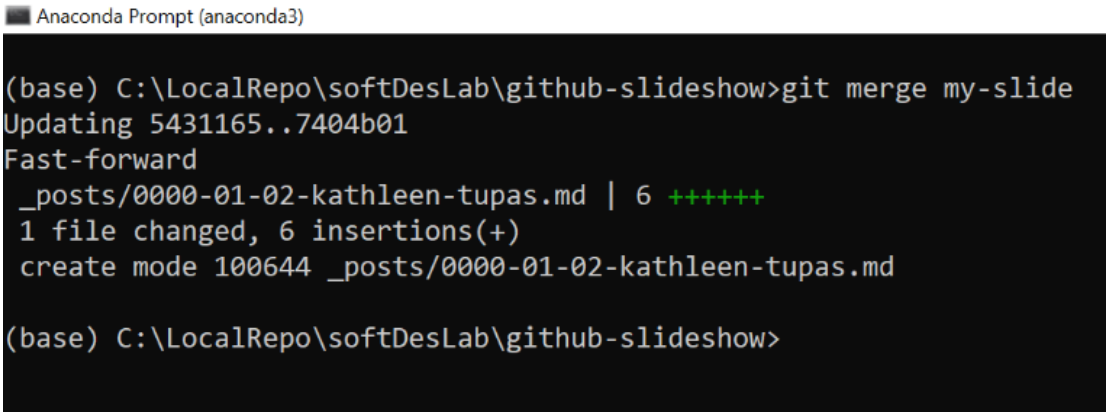


Figure 3.7 *Merging the contents of my-slide to master branch.*

After we finish all the updates necessary on my-slide branch, we merge it on the master branch so that we can integrate the updates into a single branch.

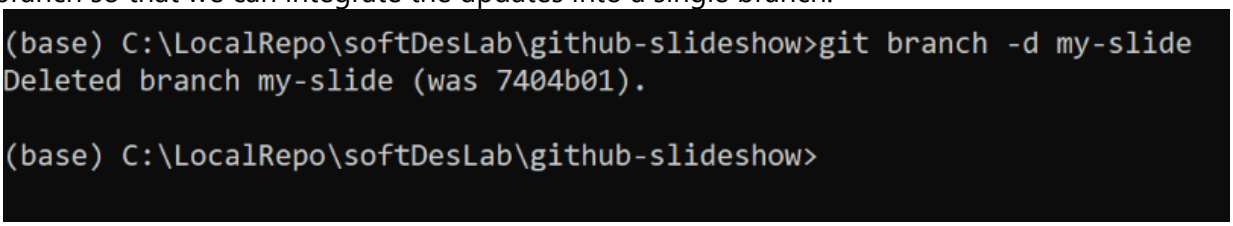


Figure 3.8 *Deleting the my-slide branch.*

Since we already merged the contents of my-slide branch to master branch, we deleted it using the command git branch -d.

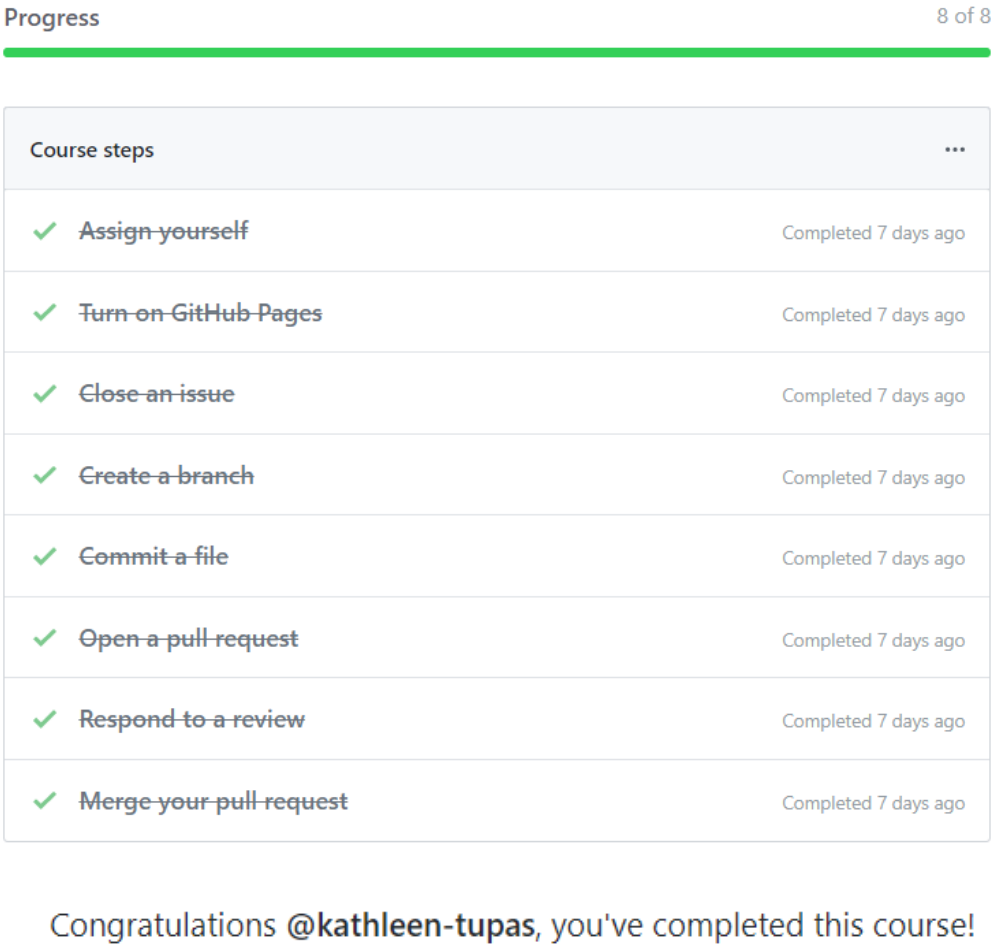


Figure 3.9 *Completed the Github Learning Lab.*

We successfully finished the tutorial of git in GitHub Learning Lab. We learned about the GitHub flow, useful commands that can be used in future projects, and how important git is as a version control system for tracking changes in our source code in software development.