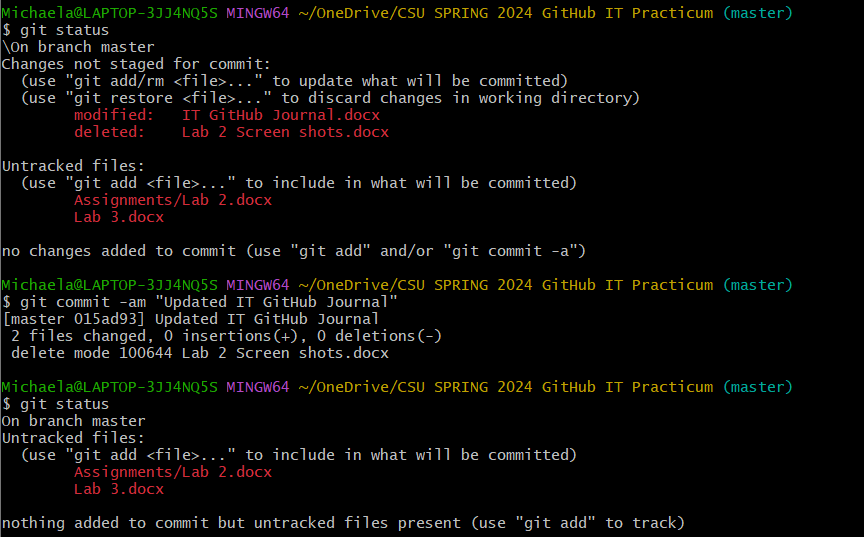
Lab 3 Screen shots

Please use the learning materials in Module 3 to complete this lab. Take up to 11 screen shots. Show the commands with the results if possible.

**Make one small change to a tracked file in your existing repo. Commit it using the git commit -am “Short commit message” method. Run git status and take a screenshot** ([Demo video here](https://youtu.be/ugp8O7mhplM)).



**If you don’t have any source code files or text files in your repo, please add one. Use the technique above to commit it. Now make a small change. Commit again. Run git log -p to see the lines that were changed. Screen shot this.**

**A screenshot of a computer program

Description automatically generated**

**Try the git show command with a commit ID. Take a screen shot.**

**A screen shot of a computer

Description automatically generated**

**Try the git log –stat command. Screen shot.**

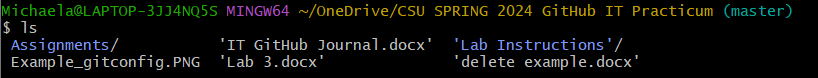
A computer screen shot of a computer program

Description automatically generated

What happens when you try **git diff -u**? Why is that?



**You can delete files in your repository using git rm. First look at the contents of the directory with ls, and take a screenshot:**

****

**then delete the file with git rm, then check the contents with ls again, and finally check the status with git status. Take a screen shot.**

**A screen shot of a computer program

Description automatically generated**

Lesson learned: we see that by calling **git rm**, the file was deleted from the directory, and the change was also staged to be committed in our next commit.

**Now is a good time to try git diff -u. Why is that? Take a screen shot.**

A black screen with yellow text

Description automatically generated

**Let's call git commit and send a message indicating that we've deleted the unneeded file. As usual, we get a bunch of stats when we do the commit. Take a screen shot.**

**A screen shot of a computer

Description automatically generated**

**You can use the git mv command to rename files in the repository. Rename one of your repo files using git mv. Now run git status and take a screen shot. The status shows us that the file was renamed and clearly displays the old and new names.**

A screen shot of a computer screen

Description automatically generated

**Add a gitignore file to our repo and edit this file: enter the name of the file(s) you want to leave out of tracking by git. Show screen shot of git status now:**

**A computer screen shot of a program

Description automatically generated**

**The .gitignore file needs to be tracked just like the rest of the files in the repo, so please track and commit it, and show screen shot of git status again.**

**A screenshot of a computer screen

Description automatically generated**