**Basic Git commands**

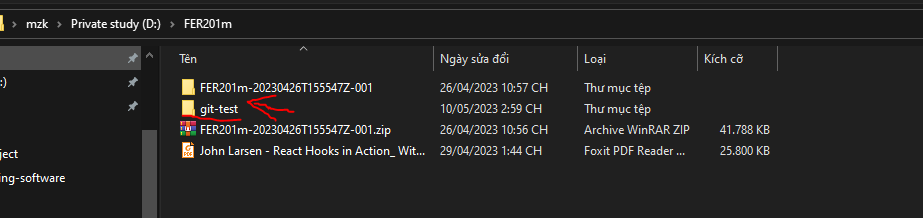
#### **Objectives and Outcomes**

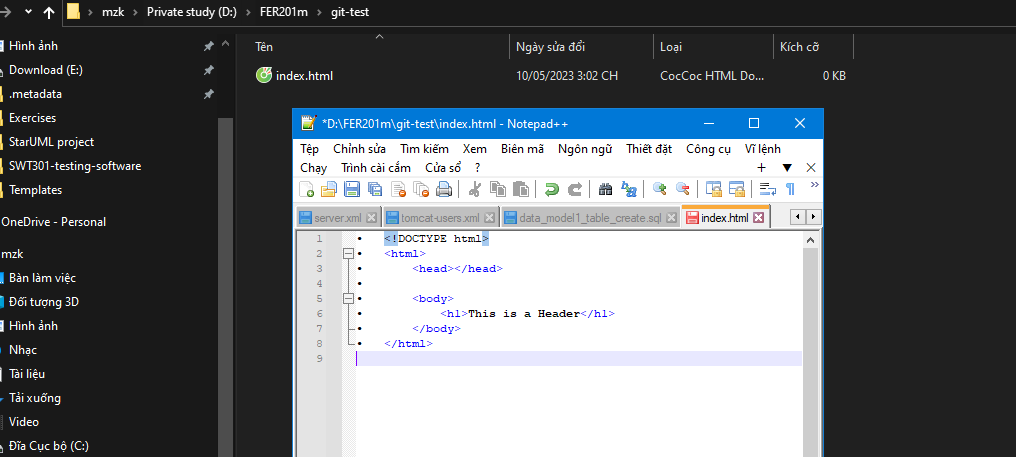
In this exercise you will get familiar with some basic Git commands. At the end of this exercise you will be able to:

* Set up a folder as a Git repository
* Perform basic Git operations on your Git repository

#### **Basic Git Commands**

* At a convenient location on your computer, create a folder named **git-test**.



* Open this git-test folder in your favorite editor.
* Add a file named *index.html* to this folder, and add the following HTML code to this file:
* <!DOCTYPE html>
* <html>
* <head></head>
* <body>
* <h1>This is a Header</h1>
* </body>
* </html>
* 

#### **Initializing the folder as a Git repository**

* Go to the git-test folder in your cmd window/terminal and type the following at the prompt to initialize the folder as a Git repository:

git init

#### **Checking your Git repository status**

* Type the following at the prompt to check your Git repository's status:

git status

**Adding files to the staging area**

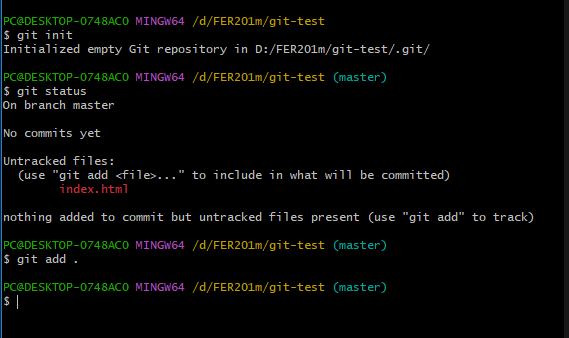
* To add files to the staging area of your Git repository, type:

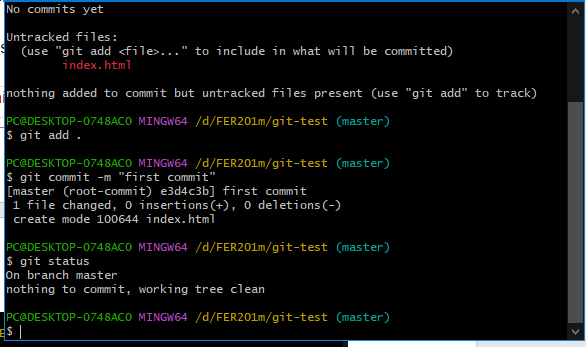
git add .

**Commiting to the Git repository**

* To commit the current staging area to your Git repository, type:

git commit -m "first commit"

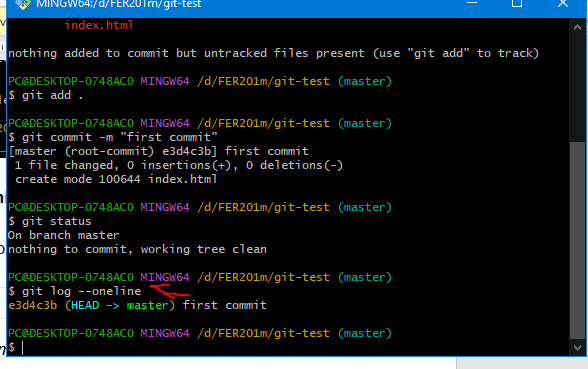


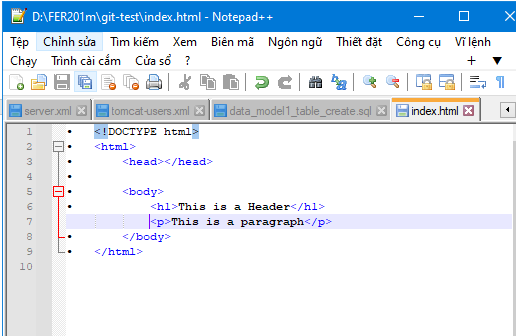


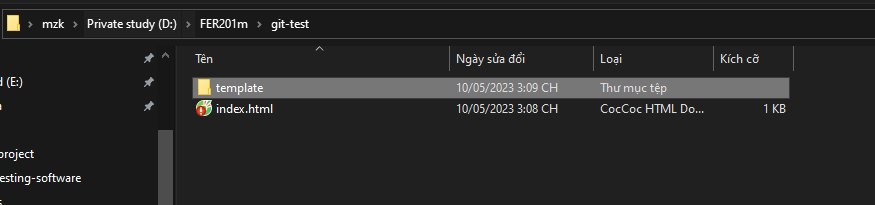
**Checking the log of Git commits**

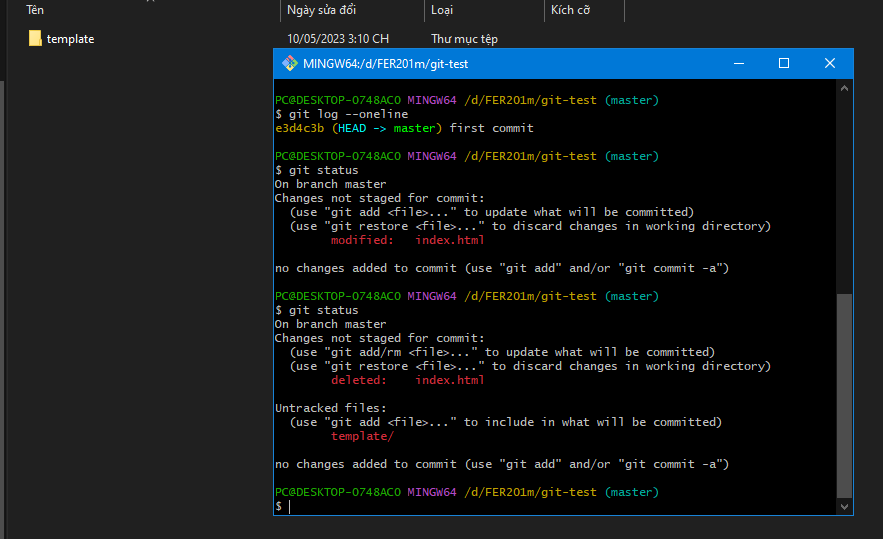
* To check the log of the commits to your Git repository, type

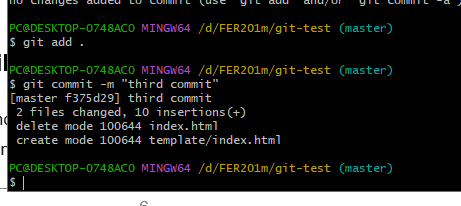
git log --oneline

* 
* Now, modify the *index.html* file as follows:
* <!DOCTYPE html>
* <html>
* <head></head>
* <body>
* <h1>This is a Header</h1>
* <p>This is a paragraph</p>
* </body>
* </html>



* Add a sub-folder named **templates** to your **git-test** folder, and then add a file named test.html to the templates folder. Then set the contents of this file to be the same as the index.html file above.
* 
* Then check the status and add all the files to the staging area.
* Then do the second commit to your repository
* Now, modify the index.html file as follows:



* <!DOCTYPE html>
* <html>
* <head></head>
* <body>
* <h1>This is a Header</h1>
* <p>This is a paragraph</p>
* <p>This is a second paragraph</p>
* </body>
* </html>
* Now add the modified index.html file to the staging area and then do a third commit.
* 

#### **Checking out a file from an earlier commit**

* To check out the index.html from the second commit, find the number of the second commit using the git log, and then type the following at the prompt:

git checkout <second commit's number> index.html

#### 

#### **Resetting the Git repository**

* To discard the effect of the previous operation and restore index.html to its state at the end of the third commit, type:

git reset HEAD index.html

* Then type the following at the prompt:

git checkout -- index.html

* You can also use git reset to reset the staging area to the last commit without disturbing the working directory.

#### **Conclusions**

At the end of this exercise you should have learnt some basic Git commands. Experiment with these commands until you fully understand how to use Git.