

Assignment (100 points)

CS418/518 – Web Programming

TASK 1: Install GIT

Git is a distributed version control system that we will use in this course to submit coding assignment and develop our respective codebases. To use Git in your system, you must first install the Git environment. Your operating system may already have Git installed. Open a terminal and type `git --version` at your command-line to verify this. If an error similar to "command not found" is returned, consult the [Git homepage \(https://git-scm.com/book/en/v2/Getting-Started-Installing-Git\)](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git) to install the program for your operating system.

TASK 2: Create two Git repositories

In this course, we will use GitHub for assignment submission from our local Git repositories. Documents in Git reside in a "repository", accessible using both the `git://` protocol and through the web interface on GitHub at <https://github.com/yourusername/yourrepositoryname>.

You should replace "yourusername" with your actual username and "yourrepositoryname" with your actual repository name. This will be the repository for your course project.

2.1 Create a repository for your course project

GitHub now allows unlimited free private Git repositories. This is necessary for the course project. For this assignment, only a standard GitHub account is needed. Create a blank, [private](#) repository ("repo" henceforth) under your account. The name of the repo can be anything **BUT** [cs418518-s24](#) (to prevent a clash for this assignment). In this repo, you will host your project as the course progresses. Add the instructor (GitHub user nasreen_arif@live.com) as a collaborator to your repo (How to [invite collaborators to a personal repository](#)).

2.2 Create a fork to the class repository

1. Fork the public Git repo at <https://github.com/nasreenarif/cs418518-s24> to your GitHub account.
2. Clone your fork to your machine
3. Create a new file in the `users` directory. The file's name should be your ODU user ID (The computer ID such as `narif`, with combinations of letters and numbers. NOT the numerical ID!). An example of the file name is `narif.txt`. The content of the file should be the URL of the repo you created in section 2.1 of this assignment, such as <https://github.com/john/testrepo>.
4. Add and commit this file to your local repo.
5. Push the update you have made to the repo on your machine to your GitHub account.

6. Submit a pull request to the public repo at <https://github.com/nasreenarif/cs418518-s24> with your addition. Be sure NOT to commit invisible files (e.g., `.DS_Store` on Mac). To do this, you can add a `.ignore` file and put the following content into it:
`.DS_Store`

Hint: You should end up of hosting two GitHub repos under your own account. One has the same name as the course and the other is different.

EXTRAS

- Practice your git!!! (I cannot emphasize this enough!)
- If you are already familiar with HTML, CSS, or JavaScript, try out the [HTML Validator \(http://validator.w3.org/\)](http://validator.w3.org/), [CSS validator \(http://jigsaw.w3.org/css-validator/\)](http://jigsaw.w3.org/css-validator/), [JSLint \(https://www.jshint.com/\)](https://www.jshint.com/), and [JSHint \(https://jshint.com/\)](https://jshint.com/). Pay particular attention to the subtle differences between the final two.
 - JSLint is a JavaScript program that looks for problems in JavaScript programs.
 - JSHint is a community-driven tool that detects errors and potential problems in JavaScript code.
- Take a look at Google's HTML/CSS style guide (<https://google.github.io/styleguide/htmlcssguide.html>).

Grading rubrics.

- Creating your personal repo (20 points)
- Sharing the personal repo with the instructor (20 points)
- Forking the course repo (20 points)
- Correctly creating your file inside the users directory (20 points)
- Pushed the update to your GitHub account (10 points)
- Submit the pull request (10 points)