Tutorial 2: Code Management and Deployment [Individual and Group Deliverable]

Learning Outcomes:

- Consider the advantages and disadvantages of a code management and deployment solution given a set of project guidelines.
- Familiarize yourself with setting up a project environment on the code management and deployment solution of your choice.
- Work in groups to judge the best possible solution given a set of project guidelines.

Instructions:

The goal of Tutorial 2 is to make sure that the infrastructure you will need this semester is working for you from the very beginning. It is much easier to deal with issues at the start of the Term than later when we need to consider functionality and dependencies when troubleshooting. For this purpose, you will be setting up a few items on GitHub and GitLab, this way should any issues arise, we can more easily address them sooner rather than later.

For this tutorial, you will be working with your group and individually to complete the following requirements:

- **For the group portion of this tutorial**, you will be creating the main repo for your group project, with each member of your group having an individual branch. Each member of your group should be able to push code to the group's main repository.

  Note: *You do not have to push any code to this repo at this moment, you are simply setting up your group repo so that you have in place for when it is needed (i.e., proposal).*

  - As will be the case for all your Tutorials, Assignments, and Project Deliverables, you will be using Github for your individual/group tutorials, assignments and project deliverables, and will be mirroring your work to your GitLab account.

    Note: *Due to the folder structure you will be using for your deliverables, it may not be possible for you to only use GitLab as your repo solution, hence the recommendation to use Github. However, GitLab will still need to be used as your code will be marked and assessed for quality on GitLab.*

  - Each member of your group is expected to create a **branch** to your group's main repository. In setting up a **main group repository** for your group, you are encouraged to test that each member of your group is able to push code from their individual branches to the group's main repository

    Note: *You are strongly encouraged to test that each member of your group can successfully push some code to the master repository. This step would be meant for testing purposes, rather than quality, as* *your group will become aware of any issues in your setup, allowing TAs to help earlier rather than later in the term.*

- You will be expected to include the URLs to your group's main project repository and corresponding individual branches in your individual **README** file for Tutorial 2.

- **For the individual work portion of this tutorial**, you will be expected to setup your (front-end) project environment on your **individual tutorial repository** (i.e., tutorial2). You are also encouraged to setup up the repositories for your assignments (e.g., assignment 1, assignment 2, ...).

__Note:__ In this step, we are looking to make sure that you are able setup your own work environment. Your overall GitLab structure for the course should be similar to the structure shown on Figure 1 (See FCS GitLab submission instructions below). You may setup one repository for all your Tutorials and one repository for all your Assignments, with folders inside these repos for each tutorial and each assignment, respectively.

- Deploy your front-end tutorial 2 environment on Netlify. Mirror your repository on GitHub to your GitLab repo as GitLab will be used when marking your code for quality.
- You will be deploying a generic tutorial (or project) environment, this way we can help make sure any issues you might encounter are resolved sooner rather than later.
- Only the initial individual tutorial 2 setup should be live on a deployment link.
- Ensure your tutorial displays something, e.g., **displaying a simple "It works" message or boilerplate code from frontend frameworks would meet the requirements for this tutorial**.
- You will be expected to include the URLs to your Tutorial 2 repository, along with your Tutorial and Assignments repositories, on your **README** file.

__Note:__ You are encouraged to include the URLs to your Tutorial and Assignments repository, so that the Markers may test these are accessible through GitLab.

Submission Guidelines:

Your tutorial must be submitted through **Brightspace (i.e., README file), GIT and be remotely accessible.**

To submit your work to Brightspace:

- Create a README.txt or README.md file, follow the guidelines specified in the README template provided through Brightspace.
  - Rename your README file to match naming conventions specified in the Course Syllabus (**T#_FName_LName_README.txt or T#_FName_LName_README.md).**
  - **Include** the URLs to your Group Project's main repository and corresponding branches, as well as the URLs to you Tutorials and Assignments repositories.

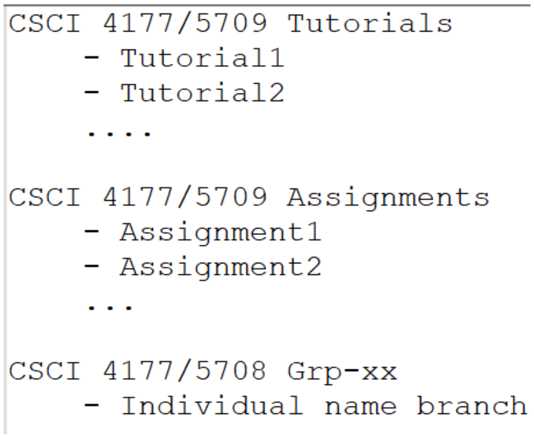
    Note: Your group repository does not need to have any code pushed to it. *Feel free to edit the README template provided on Brightspace as you see fit. Your README file should include your name, GIT repository link, deployment link, and any code references. Failure to submit a README file and/or not include a link to your deployed application will result in a grade of 0.*
  - **Include** the URLs to your Tutorial 2 repository, as well as the URL to your deployed application for Tutorial 2 (e.g., displaying an **"It works"** message or boilerplate code from frontend frameworks would suffice for this tutorial.)
- Ensure you submit your work by the **due date specified on Brightspace.**

To submit your work to FCS GitLab:

- Push your Tutorial 2 code to its repository and deploy the application to Netlify or any other deployment platform of your choice. For this purpose you will be using Github, however, you will be expected to mirror your Github repo to your GitLab repo as GitLab will be used when marking your code for quality.

  Note: *Make sure that the deployment link you include in your README file matches the deployment link for this tutorial. Your GIT repository must be individual and private and be accessible to the Instructor and Teaching Assistants.*

- Follow the folder structure requirement shown on **Figure 1**. You may setup one repository for all your Tutorials and one (separate) repository for all your Assignments, with folders in these repositories for each tutorial and each assignment, respectively.



```
CSCI 4177/5709 Tutorials
    - Tutorial1
    - Tutorial2
    ....

CSCI 4177/5709 Assignments
    - Assignment1
    - Assignment2
    ...

CSCI 4177/5708 Grp-xx
    - Individual name branch
```

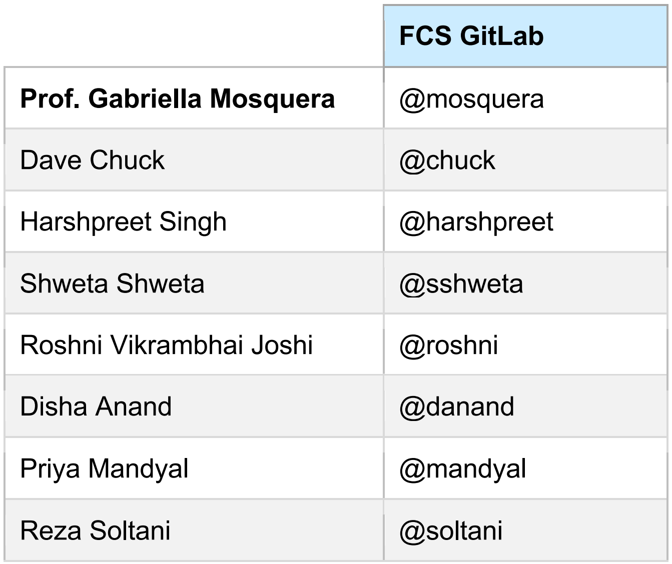
Figure 1. *GitLab Folder Structure Example.*

- **Ensure,** your repository includes a **README.txt or README.md file**, follow the guidelines specified in the README template provided through Brightspace.

Marking Rubric:

As this tutorial is a programming tutorial, the following grading criteria will be used for marking your tutorial:

- Creating a **project** group repository and corresponding branches for each group member **[2 points]**
  - **Ensure** your repos are set to **private.**
- Creating a **tutorial2** individual repository **[2 points]**
  - **Ensure** your repo is set to **private.**
- Adding the Course Instructor and Teaching Assistants as maintainers **[2 points]**



	FCS GitLab
Prof. Gabriella Mosquera	@mosquera
Dave Chuck	@chuck
Harshpreet Singh	@harshpreet
Shweta Shweta	@sshweta
Roshni Vikrambhai Joshi	@roshni
Disha Anand	@danand
Priya Mandyal	@mandyal
Reza Soltani	@soltani

- Deploying your **tutorial2 app** on **Netlify** **[4 points]**
  - **Ensure** a link to your Tutorial 2 application, individual Tutorial 2 repo, as well as your group project's main repo and corresponding branches are included in your **README** file.

Note: *Failure to include a README file as part of your tutorial deliverable on Brightspace will result in an automatic grade of* ***ZERO*** *(0).*