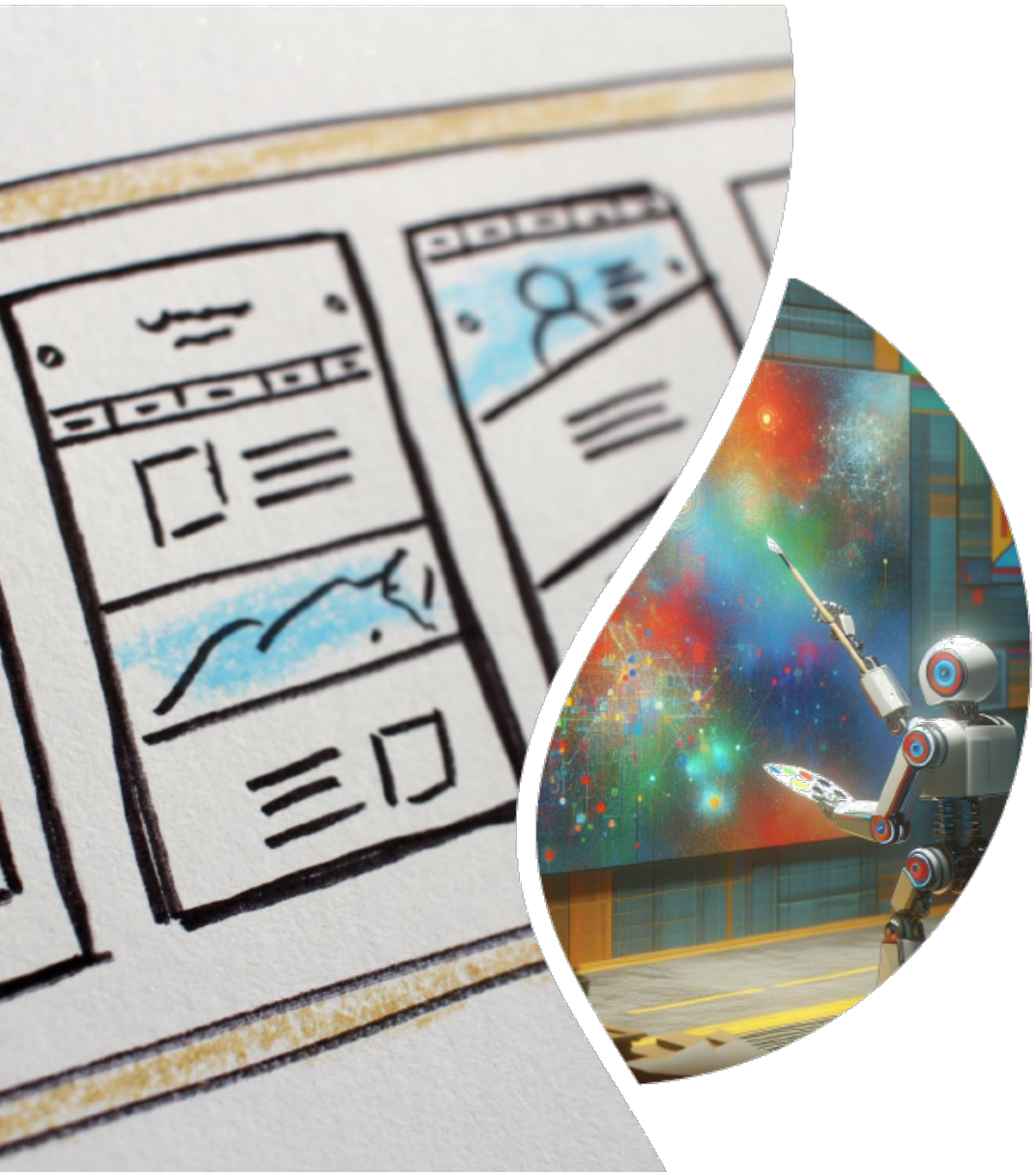


DGIN 5201
Digital Transformation
Lab 4

A2.2 and Team Meetings (CSLab 2)

Time and date:
11:35–12:25, 23-Feb-2026
Location: Mona Campbell building 1201

Image: DALL-E. Bing Image Creator. Generated by AI

Lab Overview

- Completing A2.2 — Assignment 2 part 2
  - Complete e3 and e4 on timberlea
  - Submit e1, e2, e3, e4 to course GitLab
- Team meetings

Exercise e3 Overview (Lab)

- We have a simple registration form on web in e2
- It can be printed and filled manually
- There is a link to available "material"
  - but no "material" yet
  - Let us provide material
- Login to your timberlea account as usual

Example e3: Next Iteration of Our Site: Available Material

- Let us make a copy of our e2 site
- First, go back to the directory above e2:

```
cd ~/public_html/dgin5201
```

- Use command `pwd` to check your directory
- Copy e2 to e3 as an exact copy:

```
rsync -av e2/ e3/
```

- Check the new site e3 in the browser
- `rsync` is a very useful utility for copying directory structures
  - it works locally as well as over ssh
  - it copies incrementally differences, which is important if two sites are large and mostly equal
  - it may preserve permissions if we use option `-a`

Example 3: Make material available

- Create readable and accessible ('executable') directory `material` (permissions: `rwxr-xr-x`)
- Copy PDF from: `~vlado/public/dt-mini-conf.pdf` into directory `material`
- Setup permissions for the directory `material` to be all readable and accessible (`rwxr-xr-x`), and for the file `dt-mini-conf.pdf` to be all readable (`rw-r--r--`)
- Try to access material link on the page. Does it work? Why?

Example 3: Prepare `.htaccess` in `material` directory

- Prepare file `.htaccess` and make it all readable (`rw-r--r--`):

```
Options Indexes
```

- Check material access now
- Add the following line to `.htaccess` and try accessing again:

```
Options Indexes
AddDescription "DT Conference Poster (PDF)" dt-mini-conf.pdf
```

- Add "and Information" to "DT Conference Poster" and access
- Add the following line and try again:

```
Options Indexes
IndexOptions DescriptionWidth=*
AddDescription "DT Conference Poster..." dt-mini-conf.pdf
```

- `.htaccess` file is used to configure Apache web server behaviour
  - can be used to provide a simple password-protected access

Summary of e3

- Files and permissions copied from `e2`
- File `material/dt-mini-conf.pdf` copied from a source as instructed, and permissions properly set
- File `material/.htaccess` prepared as specified

Example e4: Next Iteration of Our Site: Password Protection

- User `rsync` again to make a copy of e3 to e4
- Example 4 (e4) will be used to demonstrate password protection

Example 4: Simple Password Protection

- cd to e4 directory and let us prepare a password
- In a locally-only readable file `pw` (`rw-------`) we can save a password for our reference: `dt dt5201`
- Prepare the password for the site using the command:

```
htpasswd -bc .htpasswd dt dt5201
```

- Make the file `.htpasswd` all-readable and check its contents
- Prepare the file `.htaccess` and make it all readable:

```
AuthType Basic
AuthName dgin5201
AuthUserFile /users/webhome/<your_csid>/dgin5201/e4/.htpasswd
AuthGroupFile /dev/null
<Limit GET POST>
require user dt
</Limit>
```

- Check that site is password-protected

Summary of e4

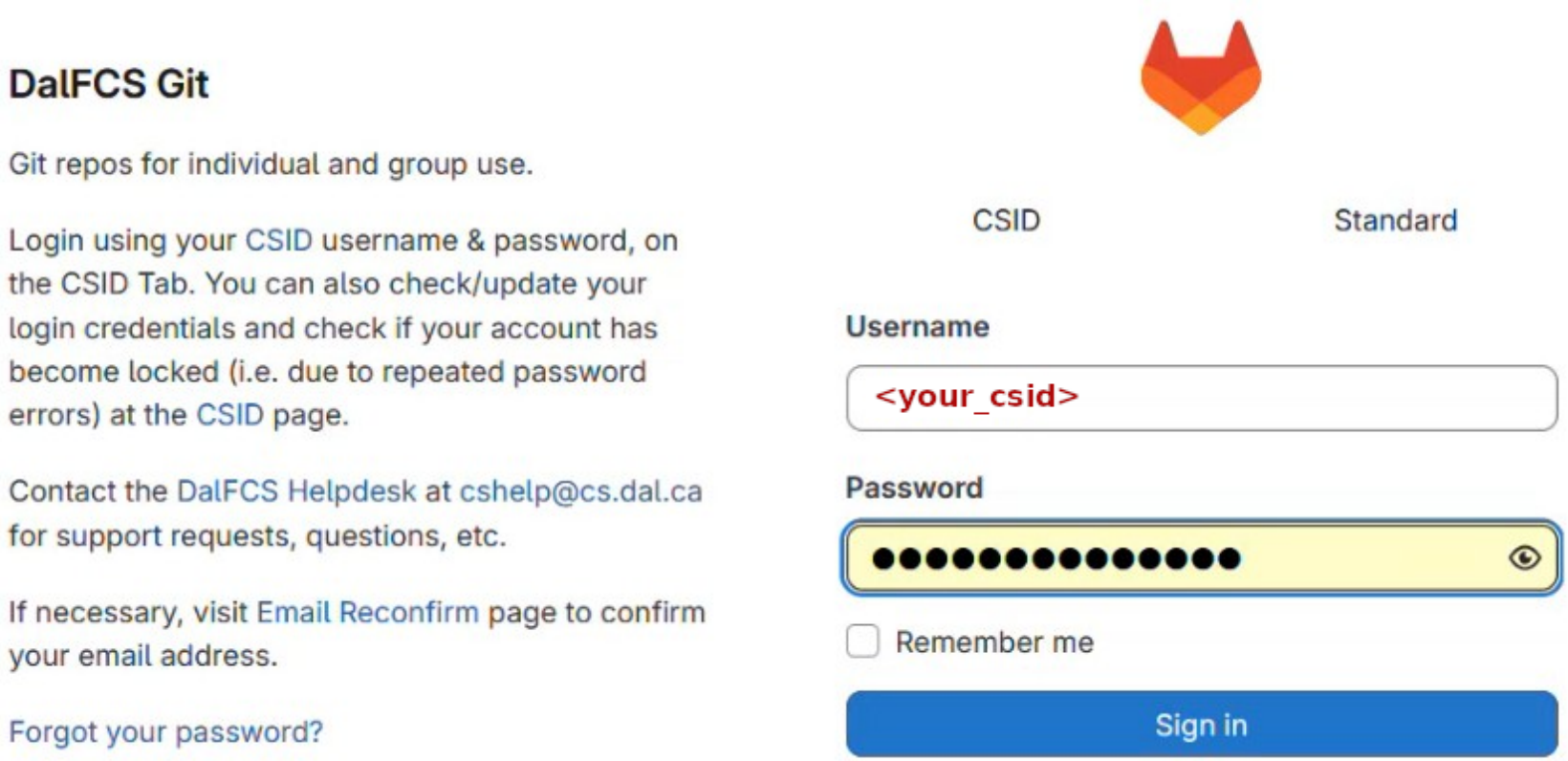
- Files and permissions copied from e3
- `pw` file with permissions `rw-------`
- `.htpasswd` file with permissions `rw-r--r--` and appropriate content set up with the `htpasswd` command
- `.htpasswd` file with permissions `rw-r--r--` and content set up for password protection as given in class

Concepts Review: Example 4

- `htpasswd` command, password saved as hash
- Using `.htaccess` for password-controlled access

Step 1. Logging into DalFCS GitLab Website

- Open your Web browser and go to: `https://git.cs.dal.ca`

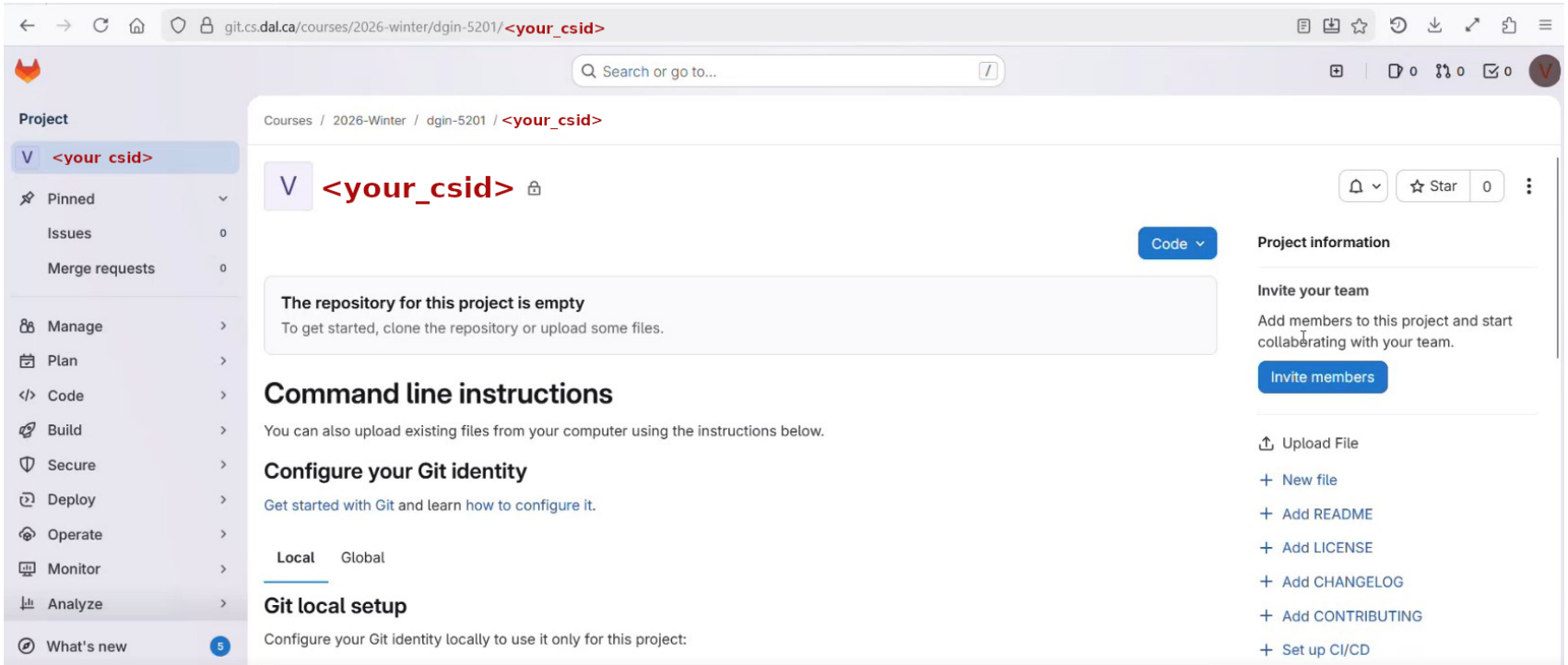


Step 2. Find your CSID Course Project in DGIN5201 Group

- The next step is to find your course project
- It is in the course group for the current term
- It is in the dgin-5201 course group
- Its name is your CSID
- There are different ways to search or navigate to the project
- You can also type directly the URL of the project as shown

Your Course GitLab Project (Repository)

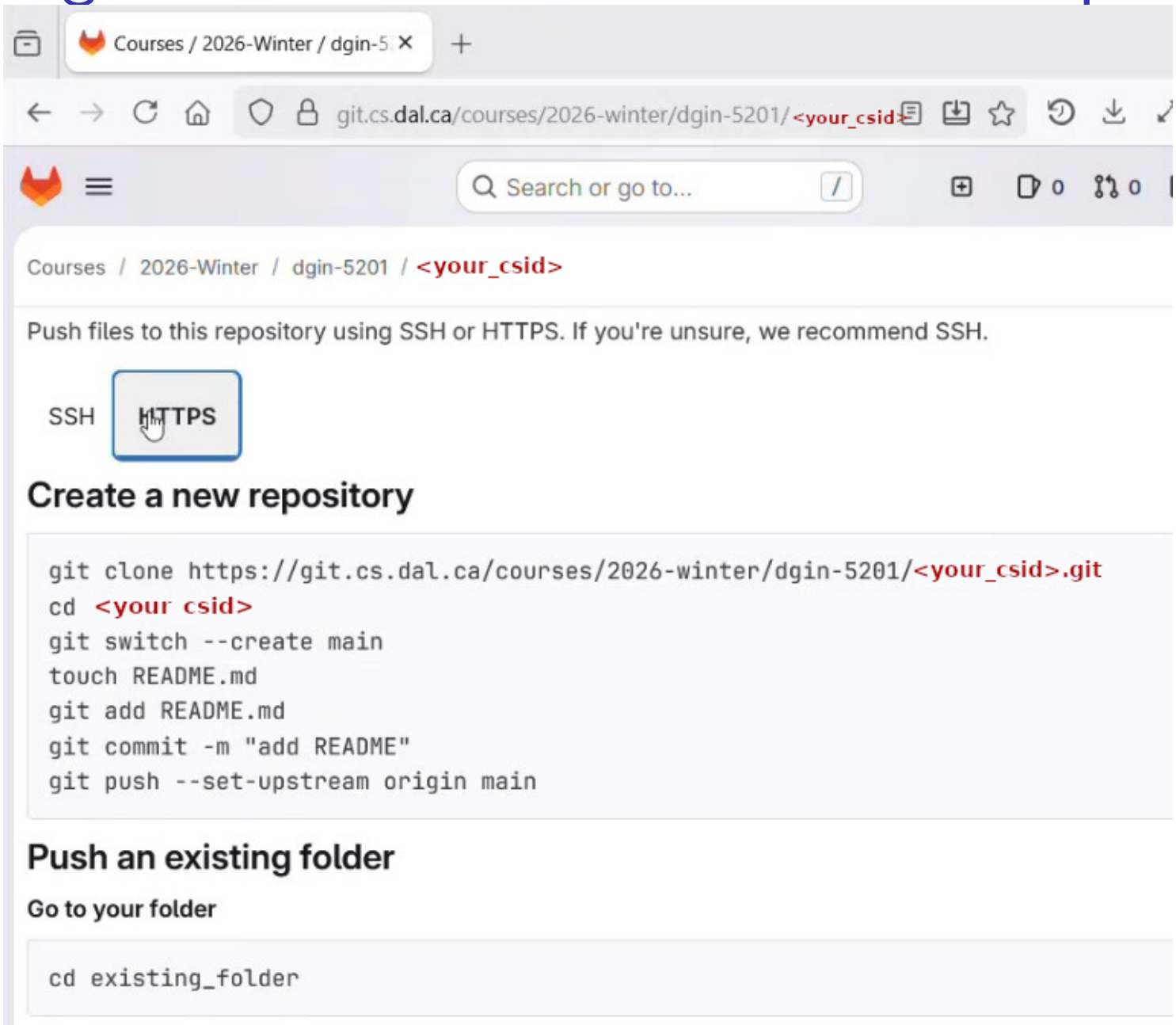
- It is named as your CSID and in the course group (2026-winter/dgin-5201/<your_csid>)
- URL: `https://git.cs.dal.ca/courses/2026-winter/dgin-5201/<your_csid>`



Step 3: Uploading your Files from `timberlea`

- In this step you should upload your lab files from `timberlea` into the GitLab server
- The instructions are shown in the GitLab page
- You should also open another command-line window for ssh login to `timberlea`

Selecting HTTPS Instructions for Initial Upload



Instructions to Upload our Files

Push an existing folder

Go to your folder

```
cd existing_folder
```

Configure the Git repository

```
git init --initial-branch=main --object-format=sha1
git remote add origin https://git.cs.dal.ca/courses/2026-winter/dgin-5201/<your_csid>.git
git add .
git commit -m "Initial commit"
git push --set-upstream origin main
```

Login to timberlea Server

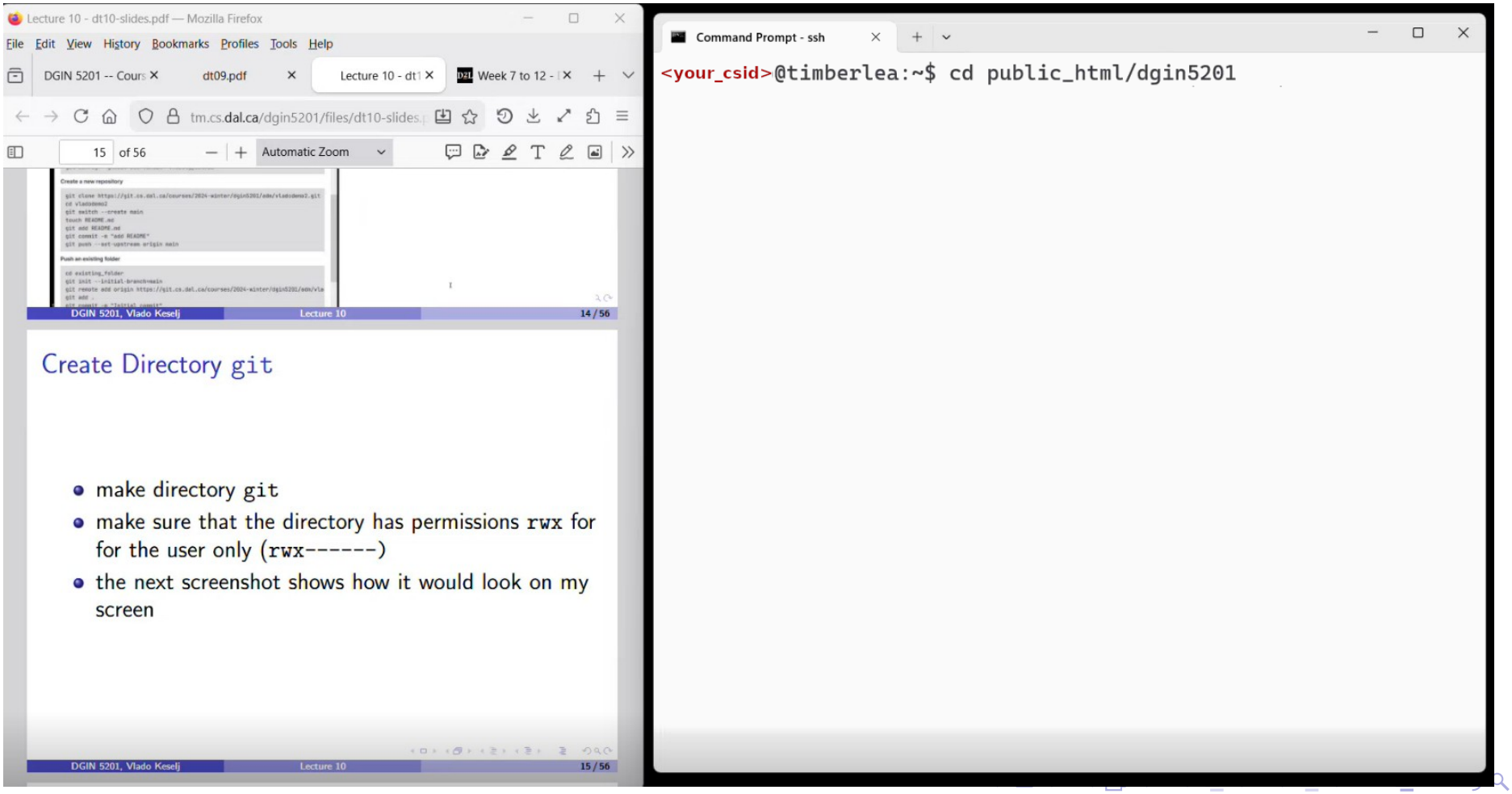
- If not logged in, we login to timberlea server
- We open a command-line window or terminal window in which we type the command:

```
ssh <your_csid>@timberlea.cs.dal.ca
```

  where instead of <your_csid> you should use your own CSID
- or maybe you can use PuTTY, mobaxterm, or other SSH application
- You should try to have two windows: the web browser with GitLab, and the command-line window

Going to your Labs Directory

- Change directory to `~/public_html/dgin5201` and list contents
- your screen with two windows could look as follows:



Create Directory `git`

- make sure that you are in directory `~/public_html/dgin5201`
- create a directory `git`
- make sure that the directory has permissions `rwx` for the user only (`rwx------`)
- the next screenshot shows how it would look on my screen


Command Prompt - ssh vlad
vladodemo@timberlea:~$ cd public_html/dgin5201
vladodemo@timberlea:~/public_html/dgin5201$ PS1='$ '
$ pwd
/users/faculty/vladodemo/public_html/dgin5201
$ ls
e1 e2 e3 e4
$ mkdir git
$ ls -la
total 0
drwx--x--x 7 vladodemo csfac 57 Feb 12 10:50 .
drwx--x--x 4 vladodemo csfac 34 Feb 3 11:06 ..
drwx--x--x 2 vladodemo csfac 43 Feb 10 09:09 e1
drwx--x--x 2 vladodemo csfac 24 Feb 5 11:23 e2
drwx--x--x 3 vladodemo csfac 40 Feb 10 10:47 e3
drwx--x--x 3 vladodemo csfac 84 Feb 10 11:13 e4
drwx------ 2 vladodemo csfac 6 Feb 12 10:50 git
$

Copy directories e1...e4 into directory git

- Copy the directories e1, e2, e3, and e4 into the directory git using the rsync commands as follows:

```
rsync -av e1/ git/e1/
rsync -av e2/ git/e2/
rsync -av e3/ git/e3/
rsync -av e4/ git/e4/
```

- Go to the directory git and check its contents

You can use highlighted commands to check your git directory

```
$ ls
e1  e2  e3  e4  git
$ cd git
$ ls
e1  e2  e3  e4
$ pwd
/users/faculty/vladodemo/public_html/dgin5201/git
$ ls -a
.   ..  e1  e2  e3  e4
$ ls -la
total 0
drwx------ 6 vladodemo csfac 46 Feb 12 10:52 .
drwx--x--x 7 vladodemo csfac 57 Feb 12 10:50 ..
drwx--x--x 2 vladodemo csfac 43 Feb 10 09:09 e1
drwx--x--x 2 vladodemo csfac 24 Feb  5 11:23 e2
drwx--x--x 3 vladodemo csfac 40 Feb 10 10:47 e3
drwx--x--x 3 vladodemo csfac 84 Feb 10 11:13 e4
$
```

Instructions to Upload our Files

- We use instructions in the GitLab page to upload exercises
- Use 5 git commands as shown, using your CSID

Push an existing folder

Go to your folder

```
cd existing_folder
```

Configure the Git repository

```
git init --initial-branch=main --object-format=sha1
git remote add origin https://git.cs.dal.ca/courses/2026-winter/dgin-5201/<your_csid>.git
git add .
git commit -m "Initial commit"
git push --set-upstream origin main
```

GitLab Upload Completed

- Go to GitLab in the browser and refresh your repository page
- Directories e1, e2, e3, and e4 should be visible
- You can browse them to check the file contents
- With this, we finished uploading e1, e2, e3, e4 to GitLab