

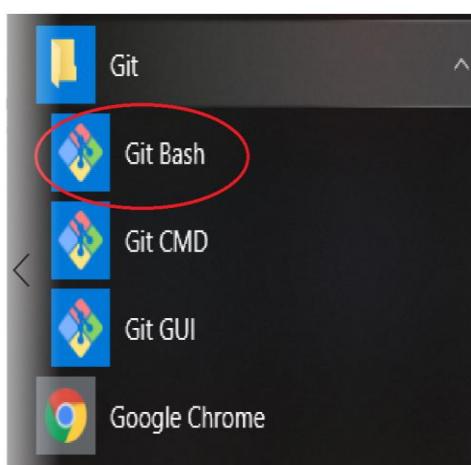
Exercice 1 : Prise en main de GitLab

Dans cet exercice, vous allez récupérer un projet sur GitLab, le modifier et le déposer dans votre propre espace GitLab (projet forked).

Suivez les instructions ci-dessous :

Installer Git

1. Pour installer GIT sur votre poste vous devez d'abord le télécharger à partir du site officiel (<https://git-scm.com/download/win>)
2. Configuration de GIT
 - Choisissez Git Bash



- Une fois l'invité ouvert, tapez les commandes suivantes pour permettre aux autres utilisateurs de pouvoir vous identifier sur les projets communs que vous aurez :

```
git config --global user.name Votre Nom Complet  
git config --global user.email votreCourriel@dti.crosemont.quebec
```

Exemple :

```
Salahudine@DESKTOP-8DL09MG MINGW64 ~  
$ git config --global user.name "Ebrahim Samer Osama"  
  
Salahudine@DESKTOP-8DL09MG MINGW64 ~  
$ git config --global user.email "1769834@crosemont.qc.ca"  
  
Salahudine@DESKTOP-8DL09MG MINGW64 ~  
c |
```

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22/site_web_Portfolio (main)  
$ git config --global user.name "Ebrahim Samer Osama"  
  
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22/site_web_Portfolio (main)  
$ git config --global user.email "sebrahim@dept-info.crosemont.quebec"
```

3. Allez sur le site <https://git.dti.crosemont.quebec> et connectez-vous avec le nom d'utilisateur et le mot de passe de votre compte du département.
4. Cliquez sur le lien sur lien Dini Ahamada/Cours Applications Web 1 A22

Dini Ahamada / Cours Applications Web 1 A22

420-G16-RO Applications Web 1

0 16 0 0 Updated 8 hours ago

Créez un nouveau projet privé à partir d'un projet public

5. Cliquer sur fork
 - Cliquez sur le bouton **Fork** en haut à droite

Cours Applications Web 1 A22

Project ID: 3024 [Leave project](#)

6 Commits 1 Branch 0 Tags 1.4 MB Project Storage

420-G16-RO Applications Web 1

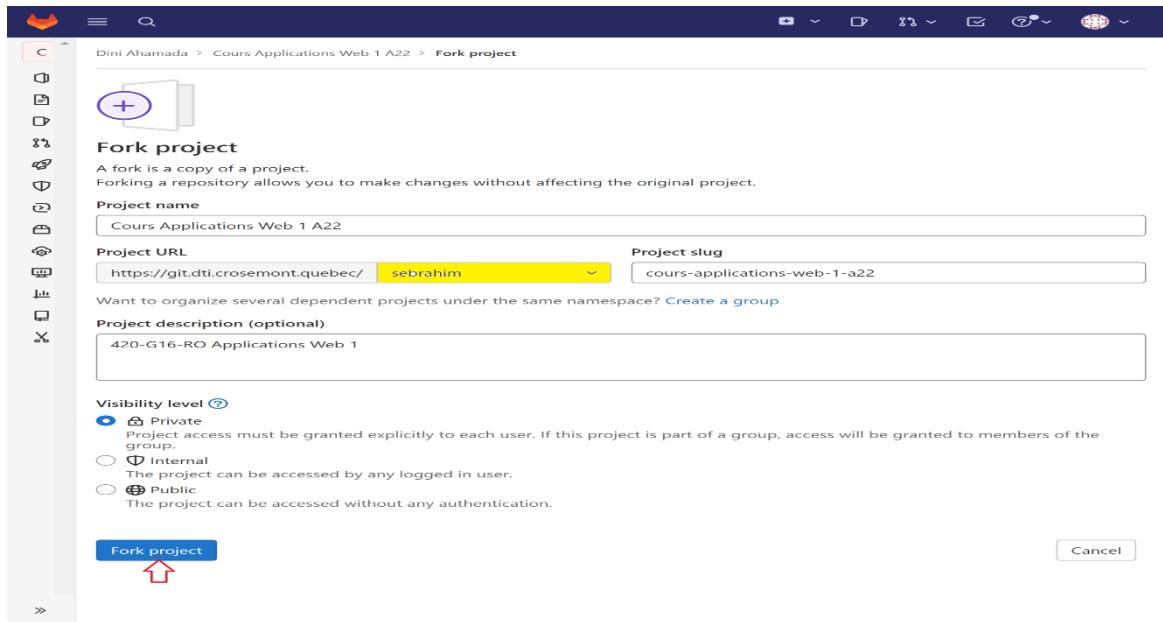
[main](#) cours-applications-web-1-a22 / + Find file Web IDE Clone

suppression copie Ahamada Dini authored 3 days ago 0616df6f

[README](#) No license. All rights reserved

Name	Last commit	Last update
Site_Web_Portfolio	Suppression du fichier cv.html	6 days ago

- Choisissez votre identité pour la destination, choisissez votre nom comme le namespace
- Cliquez Fork projet



- Vérifiez si le projet est bien copié

Samer Usama Ebrahim > Cours Applications Web 1 A22

The screenshot shows the GitHub project page for 'Cours Applications Web 1 A22'. It features a large 'C' icon, the project name, and a 'Project ID: 3052' link. Below the name are metrics: 0 commits, 1 branch, 0 tags, and 1.2 MB Project Storage. The description '420-G16-RO Applications Web 1' is present, along with 'Read more' and 'Forked from Dini Ahamada / Cours Applications Web 1 A22' links. At the top, a message says 'The project was successfully forked.' with a yellow exclamation mark icon. On the right, there are icons for a dropdown menu, stars (0), forks (0), and a 'Fork' button.

- Cliquez sur le bouton sur Enable in settings

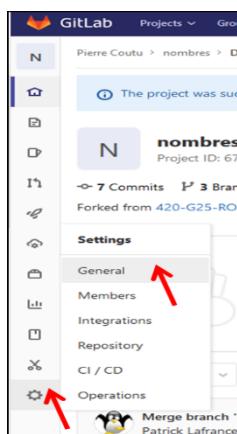
Forked from Dini Ahamada / Cours Applications Web 1 A22

The screenshot shows the GitHub Auto DevOps settings page. It includes a section titled 'Auto DevOps' with a description: 'It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.' and a 'Learn more in the Auto DevOps documentation' link. A prominent blue 'Enable in settings' button is highlighted with a red arrow. Below it, there's a checkbox for 'Default to Auto DevOps pipeline' and a note: 'The Auto DevOps pipeline runs if no alternative CI configuration file is found. Learn more.' At the bottom is a 'Save changes' button.

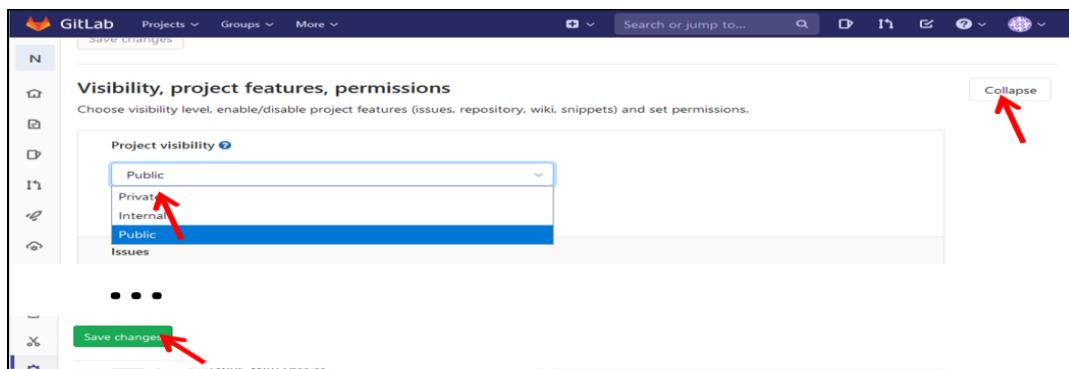
- Cliquez sur le bouton save changes

The screenshot shows the GitHub Auto DevOps settings page again, focusing on the 'Save changes' button at the bottom of the form. The button is highlighted with a red arrow.

- Choisissez le menu **Settings** (engrenage) avec la sous-option **General**



- Dans la section Visibility, project ..., cliquez à droite sur **Expand/Collapse**.
- Choisissez la visibilité **Private** et cliquez sur **Save changes** au bas.



Utiliser SSH (Secure Shell) pour l'authentification

- En ajoutant SSH, vous pouvez connecter sur GitLab sans besoin d'utiliser username et password à chaque fois
- Etape 1: Executer la commande suivante : ssh-keygen

```
dahamada@PT011374 MINGW64 ~
$ ssh-keygen
```

- Taper sur la touche « Entrée » pour garder les fichiers générer par défaut

```
dahamada@PT011374 MINGW64 ~
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/dahamada/.ssh/id_rsa): |
```

- Taper sur la touche « Entrée » pour le mot de passe vide

```
dahamada@PT011374 MINGW64 ~
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/dahamada/.ssh/id_rsa):
Enter passphrase (empty for no passphrase): |
```

- Taper sur la touche « Entrée » pour la confirmation

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/dahamada/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again: |
```

- Localiser où est généré le fichier id_rsa.pub

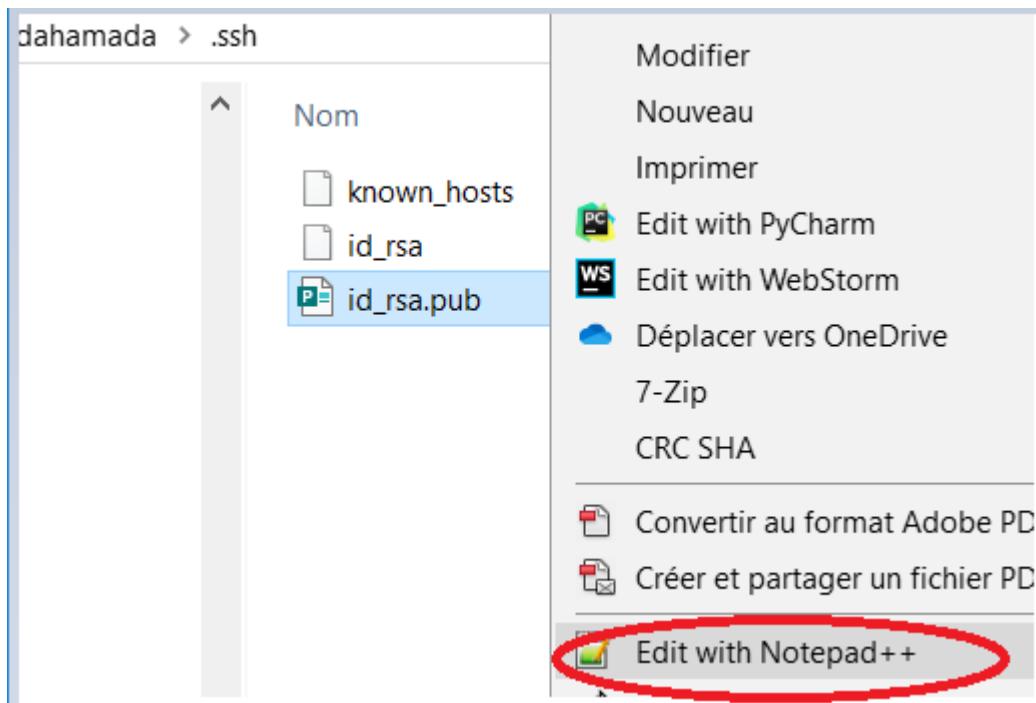
```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/dahamada/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/dahamada/.ssh/id_rsa
Your public key has been saved in /c/Users/dahamada/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:hK3SjDDTtizOgvNYSKwbqiVkzTclm+sph8/fBNlpNDU dahamada@PT011374
The key's randomart image is:
+---[RSA 3072]----+
|                   E
|                   .
|                   o   .
|                   .   .
|                   +.o .oo o
|                   =*= .o+ + o
|                   =.o=o+oSo +
|                   =+o.... . o
|                   *o=   .. .
|                   .o   oo.. o
|                   =   .   +=. .
+---[SHA256]-----+
dahamada@PT011374 MINGW64 ~
```

- Chercher le fichier .ssh sur votre poste

Windows (C:) > Users > dahamada



- Ouvrez le fichier id_rsa.pub avec un éditeur de texte « ex **Notepad++** »



- Etape 2 : Copiez le contenu de id_rsa.pub

`ssh-rsa AAAAB3NzaC1yc2EAAAQABAAgQC+xZBaKEA5Irr1GTo-NM+o3-+Bz-5h/4pRnXlcitFJpEi`

Couper
Copier →

Sur GitLab,

- cliquer sur le menu « account » et sectionner « Preferences »
- Ou bien cliquez sur le bouton « Add SSH Key »

- Cliquez sur « SSH Keys » à gauche

- Etape 3 : Ajoutez la clé que vous venez de copier
- Vous pouvez donner un titre à votre clé
- Cliquer sur le bouton « Add key »

- Étape 4 : Vérifiez que la clé SSH est bien ajoutée

SSH Key	ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQC+xZBaKEA5Irn1Slw9aWas8AwcZxfA+2afCh/4pRnXlcitfJpEeu5V9EHqirZm39Ww8wegw83OYCAhgNE3hM3jUpbxEd6l5DSzuA26t4putBX6nzKvXUNvSFnfCo+OSQ/oA0S/gHalxUeMuuW0EZ4Kdy46NgobZxAVfxzXnQ+53qT95NjtWmnwdepI7jef7xbZvOS2Na7E/AyWW4qN0XnGu6GWhXlmanH6f1AhewXB94CnROSjEnBxlXwn/qE6MrM2Pv9K
Title:	DahamadaKey
Created on:	Jun 6, 2021 10:15pm
Expires:	Never
Last used on:	Never
Fingerprints	MD5: 7f:f0:3c:30:83:96:d8:fb:15:38:fe:f4:cd:e0:da:b5 SHA256: hK35jDDTtiz0gvNYSKwbqivKzTc1m+sph8/fBNlpNDU

- Cliquer sur le menu « Projets » et puis cliquez View all projects

The screenshot shows the GitLab interface. On the left, there's a sidebar titled "Switch to" with options like "Projects", "Groups", "Milestones", "Snippets", and "Activity". Below that is a section titled "Your dashboards" with "Environments", "Operations", and "Security". A red arrow points to the "Projects" link. On the right, there's a search bar labeled "Search your projects" and a section titled "Frequently visited" which lists "Cours Applications Web 1 A22" by Dini Ahamada. At the bottom of the sidebar, there's a yellow button labeled "View all projects" with a red arrow pointing to it.

- Vérifiez que le projet contient votre nom



Cloner un projet GIT-Lab existant sur un dépôt local

- Assurez-vous que vous êtes le projet « Forked »
- Cliquez sur « clone »
- Sélectionner clone with SSH, copier l'URL

A screenshot of a GitLab project page for "Cours Applications Web 1 A22". The page shows the same details as the previous screenshot. On the right, there's a sidebar with cloning options. A red box highlights the "Clone with HTTPS" link, and another red box highlights the "Clone" button at the bottom of the sidebar. The URL "https://git.dti.crosemont.quebec/s" is also highlighted with a red box.

- Vérifiez que l'URL contient bien votre username
Exemple : git@git.dti.crosemont.quebec:sebrahim/cours-applications-web-1-a22.git
- Cloner le projet avec la commande « **git clone** »
Exemple :

```
user@DESKTOP-AQN2QR7 MINGW64 ~
$ git clone git@git.dti.crosemont.quebec:sebrahim/cours-applications-web-1-a22.git
```

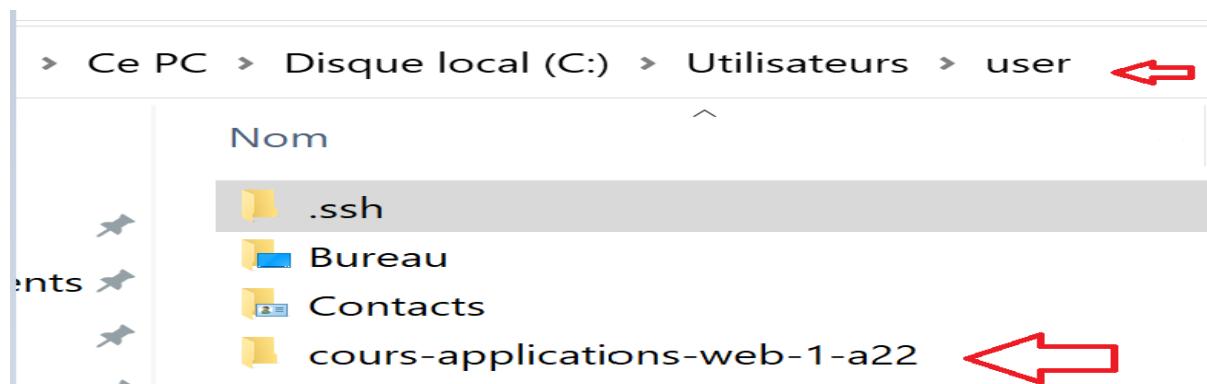
- A la question « Are you sure you want to continue connecting»
« **yes** »

```
user@DESKTOP-AQN2QR7 MINGW64 ~
$ git clone git@git.dti.crosemont.quebec:sebrahim/cours-applications-web-1-a22.git
Cloning into 'cours-applications-web-1-a22'...
The authenticity of host 'git.dti.crosemont.quebec (205.237.24.203)' can't be established.
ED25519 key fingerprint is SHA256:G5NwhAIhUSPAPj/4//qoKAsRlj+WGXFkvE4C8okIIM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

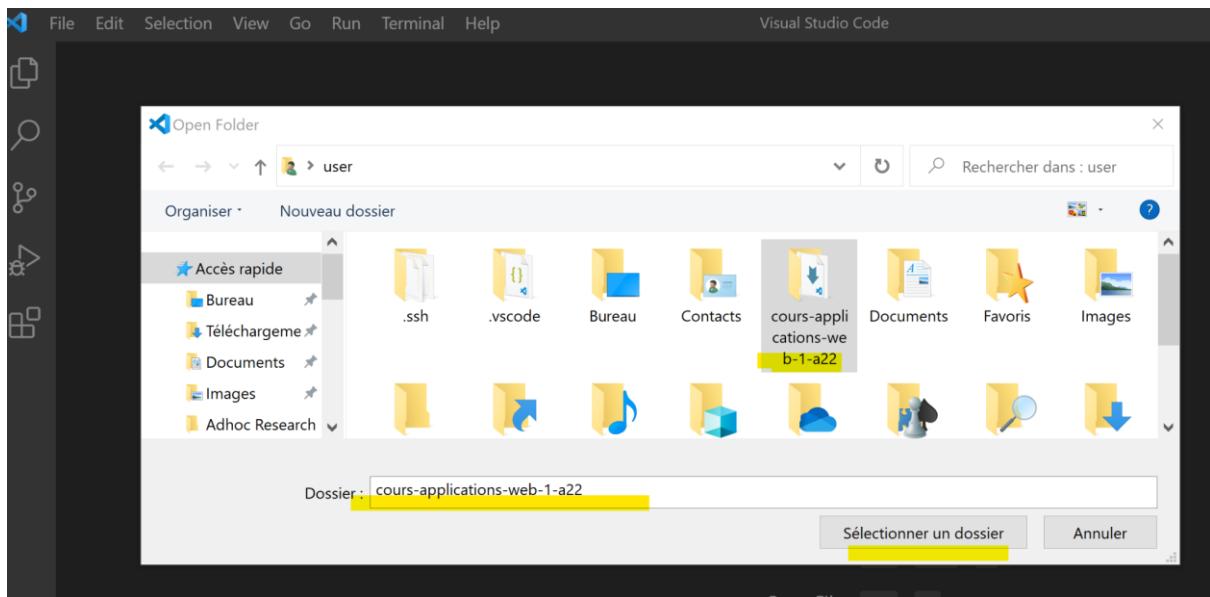
- Vérifiez le clonage c'est bien passé, vous devez voir « done »

```
user@DESKTOP-AQN2QR7 MINGW64 ~
$ git clone git@git.dti.crosemont.quebec:sebrahim/cours-applications-web-1-a22.git
Cloning into 'cours-applications-web-1-a22'...
remote: Enumerating objects: 39, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 39 (delta 9), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (39/39), 1.13 MiB | 4.65 MiB/s, done.
Resolving deltas: 100% (9/9), done.
```

- Chercher dans votre répertoire personnel, le projet cloné



- Lancez Visual Studio Code et ouvrir le répertoire



Travailler directement sur le projet forked

Supposons maintenant que nous voulons :

- Supprimer la photo : about4.jpeg
- Créer et ajouter un fichier nommé cv.html
- Modifier le fichier 3_a_propos_moi.html

```
user@DESKTOP-AQN2QR7 MINGW64 ~
$ ls
'3D Objects'
AppData/
'Application Data'@
Contacts/
Cookies@
Desktop/
Documents/
Downloads/
Favorites/
Intel/
IntelGraphicsProfiles/
Links/
'Local Settings'@
'Menu Démarrer'@
'Mes documents'@
MicrosoftEdgeBackups/
Modèles@
Music/
NTUSER.DAT
NTUSER.DAT{d04b1299-4209-11ed-900c-e12e2d56fe54}.TM.blf
NTUSER.DAT{d04b1299-4209-11ed-900c-e12e2d56fe54}.TMContainer00000000000000000000000000000001.regtrans-ms
NTUSER.DAT{d04b1299-4209-11ed-900c-e12e2d56fe54}.TMContainer00000000000000000000000000000002.regtrans-ms
OneDrive/
Pictures/
Recent@
'Saved Games'/
Searches/
SendTo@
Videos/
"Voisinage d'impression"@
'Voisinage réseau'@
cours-applications-web-1-a22/
```

- Placez dans le projet que vous venez cloner

```
user@DESKTOP-AQN2QR7 MINGW64 ~  
$ cd cours-applications-web-1-a22/
```

- Nous **voulions obtenir la dernière version** disponible à partir **du dépôt distant**. Puisque nous avons déjà une version sur notre dépôt local, on utiliser la commande **pull** au lieu de clone.

Exemple :

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22 (main)  
$ git pull origin main
```

- Si tout à jour, vous devez obtenir ceci :

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22 (main)  
$ git pull origin main  
From git.dti.crosemont.quebec:sebrahim/cours-applications-web-1-a22  
 * branch           main      -> FETCH_HEAD  
Already up to date. ←
```

- Placez dans le répertoire contenant le projet
\$ cd Site_Web_Portfolio/

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22 (main)  
$ cd Site_Web_Portfolio/
```

- Placez dans le répertoire contenant images

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22/site_Web_Portfolio (main)  
$ cd images
```

- Lancez la commande suivante pour supprimer le fichier
git **rm** about4.jpeg

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22/site_Web_Portfolio/images (main)  
$ git rm about4.jpeg
```

- Ajouter le fichier cv.html dans le projet Site_Web_Portfolio

```

EXPLORER ... cv.html U X
COURS-APPLICATIONS-... D+ E+ ⌂ ⌂
Site_Web_Portfolio > cv.html > html
> css
> document explicatif
> images
~$pllications du code.docx
1_navigation_menu.html
2_image_represenative.html
3_a_propos_moi.html M
4_service.html
5_portfolio.html
6_contact.html
7_pied.html
8_portfolio_complet.html
cv.html

```

```

Site_Web_Portfolio > cv.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Mon CV</title>
8 </head>
9 <body>
10  <h1>Mon cv</h1>
11 </body>
12 </html>

```

- Pour modifier le fichier, mettez votre nom à la place de Michel dans « 3_a_propos_moi.html » par exemple
- Vous pouvez sur Visual Studio Code le nombre des modifications faites actuellement



Après avoir fait ces changements directement sur les fichiers de notre dossier local (**zone de travail**), il faut ajouter ces changements à la **zone de préparation** de notre branche active locale avec la commande **add** suivante :

git add . (. pour tous les fichiers) ou git add nomFichier

```

user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22/site_Web_Portfolio (main)
$ git add .

```

Puis, pour enregistrer ces changements à la **branche *master* locale**, nous allons utiliser la commande **commit** :

git **commit** -m "suppression de la photo about4.jpeg, ajout du fichier cv.html, modifier le fichier 3_a_propos_moi.html"

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22/site_Web_Portfolio (main)
$ git commit -m "Suppression de la photo about4.jpeg, ajout du fichier cv.html, modifier le fichier 3_a_propos_moi.html"
```

- L'enregistrement s'est bien passé

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22/site_Web_Portfolio (main)
$ git commit -m "Suppression de la photo about4.jpeg, ajout du fichier cv.html, modifier le fichier 3_a_propos_moi.html"
[main fd0634d] suppression de la photo about4.jpeg, ajout du fichier cv.html, modifier le fichier 3_a_propos_moi.html
 3 files changed, 13 insertions(+), 1 deletion(-)
   create mode 100644 site_Web_Portfolio/cv.html
   delete mode 100644 site_Web_Portfolio/images/about4.jpeg
```

- Pour vérifier que tous les dossiers et fichiers sur le dépôt local sont à jour, on peut appliquer la commande suivante :

git **status**

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22/site_Web_Portfolio (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Pour terminer il suffit de pousser le dernier commit de la branche active sur la branche (*master*) du dépôt distant (*origin*) avec la commande **push**:

git **push** origin master/main

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22/site_Web_Portfolio (main)
$ git push origin main
```

- Le dépôt distant s'est bien passé

```
user@DESKTOP-AQN2QR7 MINGW64 ~/cours-applications-web-1-a22/site_Web_Portfolio (main)
$ git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 774 bytes | 387.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
To git.dti.crosemont.quebec:sebrahim/cours-applications-web-1-a22.git
  0616df6..fd0634d main -> main
```

- Sur GitLab vérifier sur les modifications sont bien prises en compte

Samer Usama Ebrahim > Cours Applications Web 1 A22

main cours-applications-web-1-a22 / + History Find file Web IDE Clone

Suppression de la photo about4.jpeg, ajout du fichier cv.html, modifier le... Samer Usama Ebrahim authored 7 minutes ago fd0634d3

Name	Last commit	Last update
Site_Web_Portfolio	Suppression de la photo about4.jpeg, ajout ...	7 minutes ago
README.md	Initial commit	1 week ago
README.md		

Exercice 2 : Travailler en équipe sur GitLab

Dans cet exercice vous allez travailler en équipe deux (étudiant 1, étudiant 2) et invitez et ajoutez les 2 deux professeurs (abenkhedher@dept-info.crosemont.quebec et dahamada@dept-info.crosemont.quebec) dans votre projet

Vous allez commencer par créer votre projet de Session

Essayez de créer au minimum deux pages, l'un sera fait par étudiant 1 et l'autre par l'étudiant 2. Le professeur doit être capable de voir vos modifications sur GitLab.

Inspirez-vous de l'exemple ci-dessous du projet : Album Photo.

Créer un nouveau projet directement sur le site de GIT-Lab

L'étudiant 1 fait ces différentes étapes :

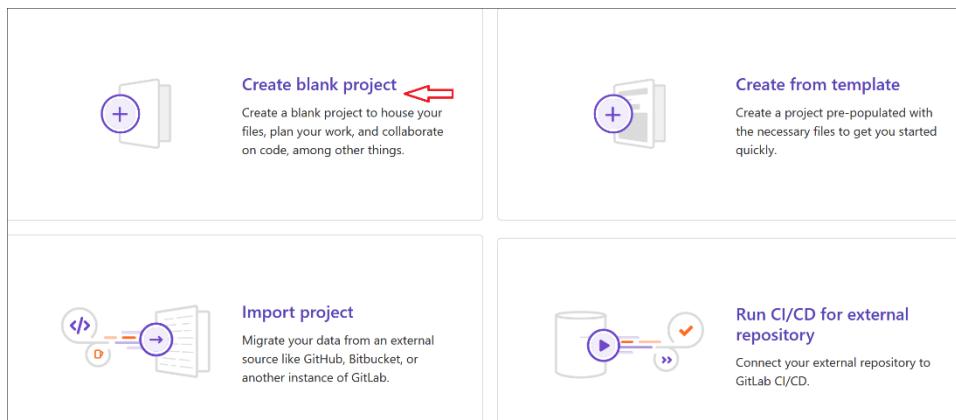
- Cliquez sur « New project »

Your projects 2 Starred projects 0 Explore projects Filter by name... Name

All Personal

C Dini Ahamada / cours-hypermedia I	Developer	420-B16-RO Techniques de développement d'applications hypermédias I classe 00298	★ 0 0 110 0	Updated 1 day ago
C Samer Usama Ebrahim / cours-hypermedia I	Maintainer	420-B16-RO Techniques de développement d'applications hypermédias I classe 00298	★ 0 0 110 0	Updated 3 hours ago

- Choisissez « Create blank project »



- Donnez un nom à votre projet
- Choisissez visibilité private
- Cliquez sur le bouton « Create projet »

New project > Create blank project

Project name

←

Project URL

Project slug

←

Want to organize several dependent projects under the same namespace? [Create a group](#).

Visibility Level (?)

Private

Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.

Internal

The project can be accessed by any logged in user except external users.

Public

The project can be accessed without any authentication.

Project Configuration

Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Enable Static Application Security Testing (SAST)

Analyze your source code for known security vulnerabilities. [Learn more](#).

Create project Cancel

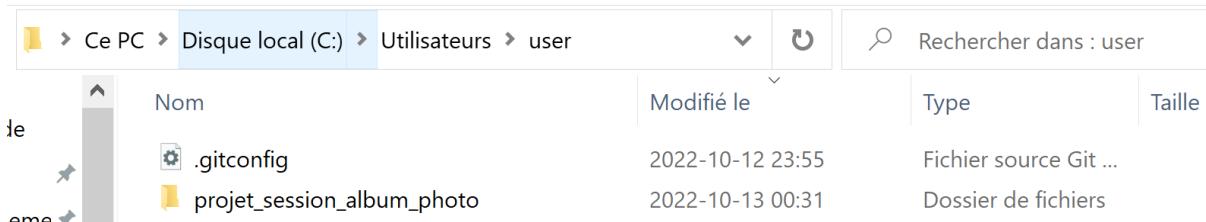
- Choisissez « members » à gauche
- Donner l'username ou addressse email de l'étudiant
- Cliquez « invite »

Créer un nouveau projet local et le pousser sur un dépôt distant

Créer un répertoire dans l'espace utilisateur, nommé le par exemple

« projet_session_album_photo »

A partir du répertoire projet_session_album_photo



Placez dans le répertoire

cd projet_session_album_photo/

```
user@DESKTOP-AQN2QR7 MINGW64 ~
$ cd projet_session_album_photo/
```

Tapez la commande suivante pour créer un nouveau référentiel/dépôt Git localement.

git init

```
:salahudine@DESKTOP-8DL09MG MINGW64 ~/labGitLab
$ git init
```

```
user@DESKTOP-AQN2QR7 MINGW64 ~/projet_session_album_photo
$ git init
Initialized empty Git repository in c:/Users/user/projet_session_album_photo/.git/
```

Copier l'url du projet que vous venez de créer

Tapez la commande suivante pour ajouter un nouveau dépôt distant Git

```
git remote add origin git@git.dti.crosemont.quebec:sebrahim/album-photo.git
```

Exemple :

```
user@DESKTOP-AQN2QR7 MINGW64 ~/projet_session_album_photo (master)
$ git remote add origin git@git.dti.crosemont.quebec:sebrahim/album-photo.git
```

Créer un répertoire « images » avec des photos dans votre projet local

Tapez la commande pour ajouter ces changements à la **zone de préparation**

```
git add .
```

```
user@DESKTOP-AQN2QR7 MINGW64 ~/projet_session_album_photo (master)
$ git add .
```

Tapez la commande suivante pour enregistrer ces changements localement avec commentaire :

```
git commit -m "ajout du répertoire"
```

```
user@DESKTOP-AQN2QR7 MINGW64 ~/projet_session_album_photo (master)
$ git commit -m "ajout du répertoire images"
```

Tapez la commande suivante :

```
git push -u origin master
```

```
user@DESKTOP-AQN2QR7 MINGW64 ~/projet_session_album_photo (master)
$ git push -u origin master
```

Créer un merge request

Samer Usama Ebrahim > Album photo

 You pushed to [master](#) at [Samer Usama Ebrahim / Album photo](#) 2 minutes ago

[Create merge request](#)



Album photo 

Project ID: 3053 

Vérifiez que votre projet local est bien déposé sur GitLab

The screenshot shows a GitLab project page for 'Album photo'. At the top, there's a header with a logo, the project name, a 'Project ID: 3053' link, and various status indicators like '3 Commits', '1 Branch', '0 Tags', and '195 KB Project Storage'. Below the header is a navigation bar with 'main' selected, a search bar, and buttons for 'Find file', 'Web IDE', and download. A merge commit from 'master' into 'main' by 'Samer Usama Ebrahim' is displayed with a timestamp of 'just now' and a commit ID of '2833963c'. Below the commit are several project management buttons: 'README', 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Enable Auto DevOps', 'Add Kubernetes cluster', 'Set up CI/CD', and 'Configure Integrations'. A table below lists a single commit with details: 'Name' (images), 'Last commit' (ajout du repertoire images), and 'Last updated' (18 minute).

Créer une branche

L'étudiant 2 fait ces différentes étapes :

```
git clone git@git.dti.crosemont.quebec:sebrahim/album-photo.git
```

```
MINGW64:/c/Users/dahamada
dahamada@PT016327 MINGW64 ~
$ git clone git@git.dti.crosemont.quebec:sebrahim/album-photo.git
```

Placez dans le répertoire généré :

```
dahamada@PT016327 MINGW64 ~
$ cd album-photo/
```

Tapez cette commande pour créer une branche :

```
git branch nomBrancheVue
```

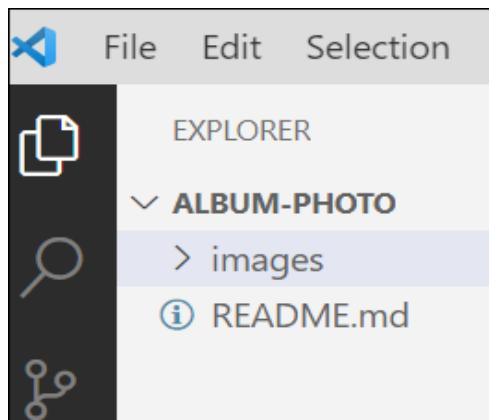
```
dahamada@PT016327 MINGW64 ~/album-photo (main)
$ git branch nomBrancheVue
```

Switchez sur la nouvelle branche pour travailler

```
git checkout nomBrancheVue
```

```
dahamada@PT016327 MINGW64 ~/album-photo (main)
$ git checkout nomBrancheVue
```

Ouvrez le projet, vous devez retrouver les modifications de votre coéquipiers



Créer un nouveau répertoire par exemple « vues » et ajouter un fichier « pageAccueil » par exemple

```
EXPLORER ... <> pageAccueil.html U X
ALBUM-PHOTO
  > images
  <> vues ●
    <> pageAccueil.html U
  README.md

<> pageAccueil.html U X
vues <> pageAccueil.html > ↗ html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta http-equiv="X-UA-Compatible" content="IE=edge">
7    <meta name="viewport" content="width=device-width, initial-scale=1.0">
8    <title>pageAccueil</title>
9  </head>
10
11 <body>
12   <h1>Notre page d'accueil</h1>
13
14 </body>
15
16 </html>
```

Ajouter le répertoire dans la zone de préparation

```
dahamada@PT016327 MINGW64 ~/album-photo (nomBrancheVue)
$ git add vues
```

Enregistrer les changements à la **branche nomBrancheVue locale**

```
dahamada@PT016327 MINGW64 ~/album-photo (nomBrancheVue)
$ git commit -m "creation du repertoire vues et ajout du fichier pageAccueil"
```

Poussez la branche au dépôt distant

```
dahamada@PT016327 MINGW64 ~/album-photo (nomBrancheVue)
$ git push origin nomBrancheVue
```

Créer un merge request

Samer Usama Ebrahim > Album photo

You pushed to **nomBrancheVue** at **Samer Usama Ebrahim / Album photo** 1 minute ago

Create merge request

A **Album photo** Project ID: 3053

3 Commits 2 Branches 0 Tags 225 KB Project Storage

Star 0 Fork 0

Commenter vos modifications

Assigner à l'étudiant 1 pour vérifier et accepter la merge

Supports **Markdown**. For **quick actions**, type **/**.

Assignees: Samer Usama Ebrahim

Reviewers: Samer Usama Ebrahim

Approvals are optional.
Approval rules

Milestone: Milestone

Labels: Labels

Merge request dependencies: Enter merge request URLs or references (e.g. path/to/project#merge_request_id)

ist the merge requests that must be merged before this one.

Merge options:

- Delete source branch when merge request is accepted.
- Squash commits when merge request is accepted.

Create merge request **Cancel**

Commits 1 **Changes** 0

13 Oct, 2022 1 commit

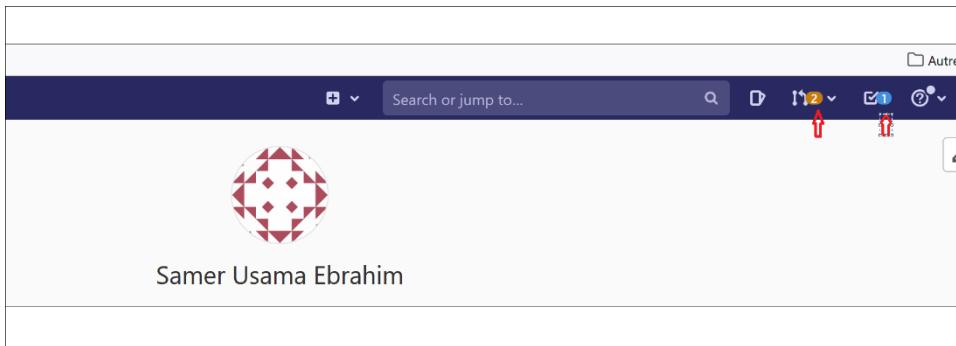
ajout du repertoire images
Samer Usama Ebrahim authored 14 minutes ago

006582f0

L'étudiant 1 doit accepter la merge :

Vous devez trouvez la requête à droite voir ci-dessous,

Cliquez sur le numéro



Cliquez open et puis le commentaire concernant la modification

A screenshot of a "Merge requests" page. At the top, there's a navigation bar with tabs: "Open" (highlighted with a red arrow), "Merged" (0), "Closed" (0), and "All" (1). Below the navigation bar is a search bar with the text "Reviewer = Samer Usama Ebrahim". The main area shows a single merge request card. The card has the title "Completer la fonction stopCountdown" and the subtitle "sebrahimi@eu-de-math!1 · created 12 minutes ago by Dini Ahamada". To the right of the card, there are icons for a person, a gear, and a file, followed by the text "0 updated 12 minutes ago".

Puis cliquez sur le bouton « Merge ». Si vous voulez refuser le merge cliquez sur « Close merge request »

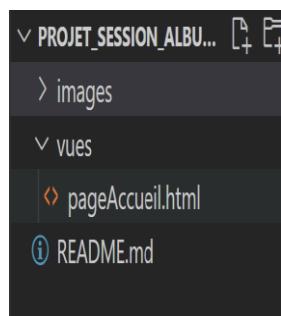
The screenshot shows a GitLab merge request page for a project named 'testbanquepc'. The merge request was opened by Pierre Coutu and is titled 'Rev du 16 mars de Pierre'. It has 0 commits and 1 change. The main panel shows a 'Request to merge devPierre into master'. There are two prominent red arrows: one pointing to the green 'Merge' button and another pointing to the orange 'Close merge request' button. The right sidebar contains fields for 'Assignee' (Pierre Coutu), 'Milestone' (None), 'Time tracking' (No estimate or time spent), 'Labels' (None), 'Lock merge request' (Unlocked), '1 participant' (Pierre Coutu), and 'Notifications' (checked). A reference number 'Reference: pcoutu/testbanquepc1' is also visible.

- Pour **obtenir la dernière version** disponible à partir **du dépôt distant**. Puisque nous avons déjà une version sur notre dépôt local, on utiliser la commande ***pull*** au lieu de clone.

Exemple :

```
dahamada@PT016327 MINGW64 ~/album-photo (main)
$ git pull origin main
```

Vous devez retrouver les modifications faites par votre coéquipier sur Visual Studio Code



Gestion des conflits

- L'étudiant 1 se place à la branche principale, en tapant la commande suivante : git checkout main

```
dahamada@PT016327 MINGW64 ~/album-photo (nomBrancheVue)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Modifier la « pageAccueil », par exemple ajouter une liste des photos

```
pageAccueil.html M X
es > ⌂ pageAccueil.html > ⌂ html > ⌂ head > ⌂ meta
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>pageAccueil</title>
9  </head>
10
11 <body>
12     <h1>Notre page d'accueil</h1>
13     <h2>Notre liste de photo</h2>
14     <ul>
15         <li>Photo1</li>
16         <li>Photo2</li>
17         <li>Photo3</li>
18     </ul>
19 </body>
20
21 </html>
```

Vérifie les modifications à l'aide de la commande git status

```
dahamada@PT016327 MINGW64 ~/album-photo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
            modified:   vues/pageAccueil.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Ajouter les modifications dans la zone de préparation

```
dahamada@PT016327 MINGW64 ~/album-photo (main)
$ git add vues/pageAccueil.html
```

Enregistrer les modifications à la **branche main locale**

```
dahamada@PT016327 MINGW64 ~/album-photo (main)
$ git commit -m "Ajoute de listes photos"
```

Pousser le changement au dépôt à distance

```
dahamada@PT016327 MINGW64 ~/album-photo (main)
$ git push origin main
To git.dti.crosemont.quebec:sebrahim/album-photo.git
 ! [rejected]          main -> main (fetch first)
error: failed to push some refs to 'git.dti.crosemont.quebec:sebrahim/album-photo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

On est dans un cas de **conflit**, en effet le fichier pageAccueil a été modifié par deux personnes différentes

Sur Visual Studio Code, voir les conflits

```
<h1>Notre page d'accueil</h1>
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<< HEAD (Current Change)
<h2>Notre liste de photo</h2>
<ul>
    <li>Photo1</li>
    <li>Photo2</li>
    <li>Photo3</li>
</ul>
=====
<p>Notre site permet de gérer des photos</p>
>>>>> 037b92d12183d35d93a1715a8c4ba8cd85fa4958 (Incoming Change)
</body>
```

Nous pouvons voir certains ajouts étranges.

• <<<<<HEAD

• =====

• >>>>> 03769d ... (incoming change)

Considérez ces nouvelles lignes comme des « séparateurs de conflit ». La ligne ===== est le « centre » du conflit. Tout le contenu entre le centre et la ligne <<<<< HEAD dans la branche principale actuelle vers laquelle pointe la réf HEAD. Autrement, tout le contenu entre le centre et >>>>> 03769d... est présent dans notre branche de merge.

On peut cliquez sur « Compare Change » pour voir les changements

```
pageAccueil.html !    <> pageAccueil.html: Current Changes ↔ Incoming Changes × ↑ ↓ ↑ ↓ ¶
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=
7      <meta name="viewport" content="width=device-wid
8      <title>pageAccueil</title>
9  </head>
10
11 <body>
12     <h1>Notre page d'accueil</h1>
13     <h2>Notre liste de photo</h2>
14     <ul>
15         <li>Photo1</li>
16         <li>Photo2</li>
17         <li>Photo3</li>
18     </ul>
19 </body>
20
21 </html>
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=
7      <meta name="viewport" content="width=device-wi
8      <title>pageAccueil</title>
9  </head>
10
11 <body>
12     <h1>Notre page d'accueil</h1>
13+     <p>Notre site permet de gérer des photos</p>
14+
15 </body>
16
17 </html>
```

Pour résoudre, il faut prendre intelligemment une décision selon le contexte soit garder la version dans notre branche principale actuelle, en cliquant sur « Accept Current change » ou choisir la version venant de la branche de merge, en cliquant sur « Accept Incoming Change », garder les 2 versions

Dans notre cas, nous avons besoin des 2 versions.

Cliquez sur « Accept both change »

```

File Edit Selection View Go Run Terminal Help
pageAccueil.html - album-photo - Visual Studio Code
EXPLORER ALBUM-PHOTO
    > images
    < vues
        < pageAccueil.html !
    README.md
... pageAccueil.html ! ...
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>pageAccueil</title>
</head>
<body>
    <h1>Notre page d'accueil</h1>
    <h2>Notre liste de photo</h2>
    <ul>
        <li>Photo1</li>
        <li>Photo2</li>
        <li>Photo3</li>
    </ul>
    =====
    <p>Notre site permet de gérer des photos</p>
<><><> 037b92d12183d35d93a1715a8c4ba8cd85fa4958 (Incoming Change)
</body>
</html>

```

Resolve in Merge Editor

Une fois le fichier modifié, pour stager le nouveau contenu mergé dans la zone de préparation, en exécutant la commande git add

```
dahamada@PT016327 MINGW64 ~/album-photo (main|MERGING)
$ git add vues/pageAccueil.html |
```

Enregistrer les changements dans la branche main locale, pour finaliser le merge/fusion, en exécutant un commit :

```
dahamada@PT016327 MINGW64 ~/album-photo (main|MERGING)
$ git commit -m "fusion mes modifications et celles mon coéquipier" |
```

Poussez le changement au dépôt distant pour la fusion

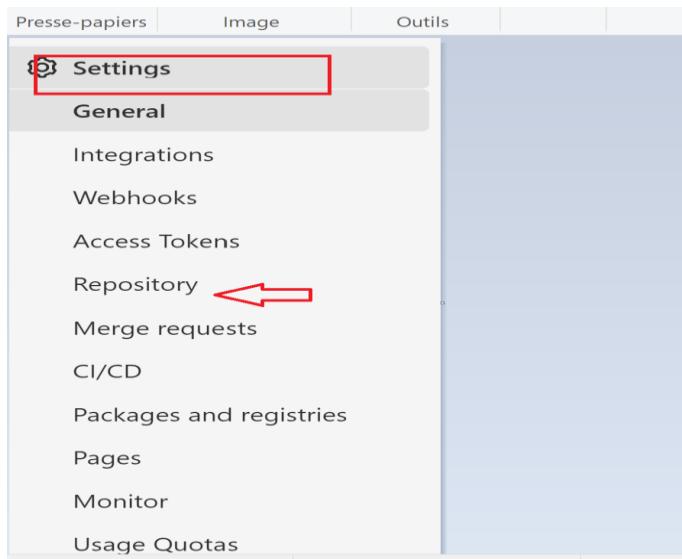
```
dahamada@PT016327 MINGW64 ~/album-photo (main|MERGING)
$ git push origin main |
```

Voici un problème éventuel que vous pouvez rencontrer lorsque vous essayez de pousser le changement au dépôt distant.

```
dahamada@PT016327 MINGW64 ~/album-photo (main)
$ git push origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 772 bytes | 772.00 KiB/s, done.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0
remote: GitLab: You are not allowed to push code to protected branches on this project.
To git.dti.crosemont.quebec:sebrahim/album-photo.git
 ! [remote rejected] main -> main (pre-receive hook declined)
error: failed to push some refs to 'git.dti.crosemont.quebec:sebrahim/album-photo.git'
```

Pour résoudre ce problème, il faut enlever la protection du projet, en suivant les étapes ci-dessous

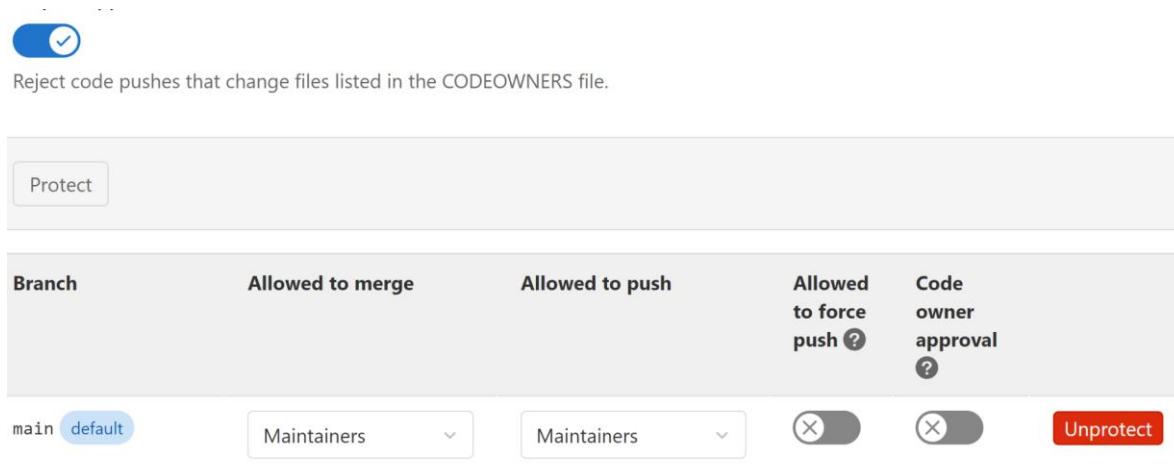
Cliquez sur « Settings » et puis « Repository »



Cliquez sur « Expand »



Cliquez sur le bouton « Unprotect »



Cliquez sur le bouton « Unprotect branch »

Branch will be writable for developers. Are you sure?



Pousser maintenant sur le dépôt à distant

```
dahamada@PT016327 MINGW64 ~/album-photo (main)
$ git push origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 772 bytes | 386.00 KiB/s, done.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0
To git.dti.crosemont.quebec:sebrahim/album-photo.git
  037b92d..272dca7  main -> main
```

Vous pouvez voir que les modifications sont prises en charges

 fusion mes modifications et celles mon coéquipier Ahamada Dini authored 14 minutes ago	272dca7c 
<hr/>	
Name	Last commit
..	
 pageAccueil.html	fusion mes modifications et celles mon coéquipier 14 minutes ago