



FACULTE DES SCIENCES ET TECHNOLOGIES

TD N° 1 : Système

Blomy ANTOINE

L3, Sciences Informatiques

Sous la direction du professeur :

Ismaël SAINT-AMOUR

20 octobre 2025

Objectif de ce Travail Dirigée(TD)

Dans ce travail dirigé, je vais apprendre comment utiliser Git ou Microsoft PowerShell dans la gestion, le **suivi** et la collaboration de projets informatiques à l'aide de ces systèmes de contrôle de version moderne.

Par ailleurs, je dois installer Git sur mon ordinateur, contrôler quelle version qui est installer en faisant dans le Terminal « git - -version ». Puis je dois créer un compte en ligne sur GitHub. Par la suite je vais reproduire le même TD que le professeur a vu avec nous en classe.

De plus, je vais découvrir l'utilisation de GitHub et créer un nouveau dépôt sur GitHub au nom de « TD », le cloner localement par la suite. Je vais créer dans le dossier TD des sous-dossier.

Après toutes ces démarches, je dois héberger le TD en ligne sur GitHub et déposer le lien sur Classroom pour le professeur.

Description du travail Effectué

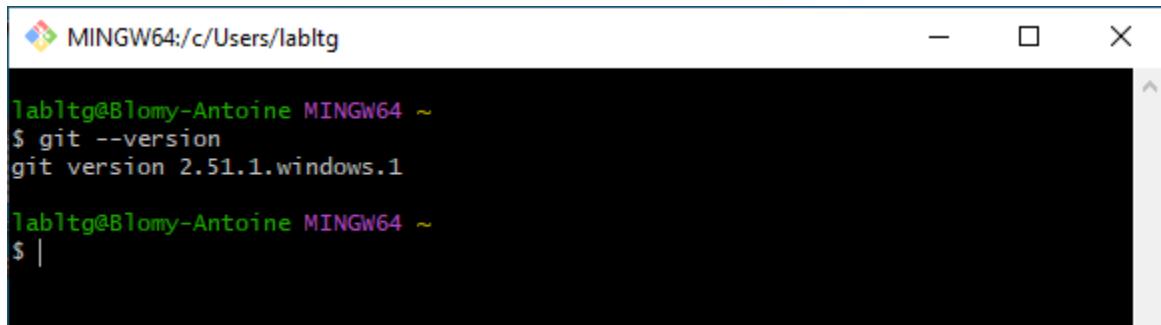
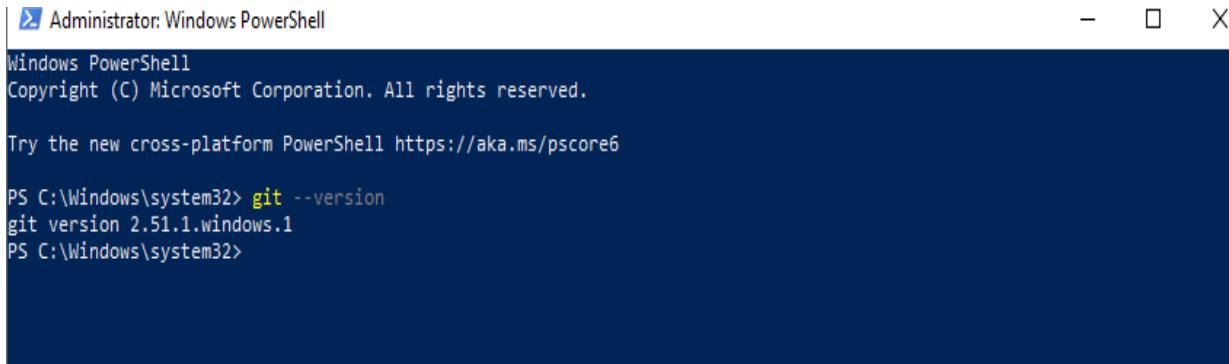
Ce TD m'a permis de découvrir les différentes pour installer et configurer Git, créer un compte GitHub et relier mon dépôt local à un dépôt distant.

1- Installation De Git Sur Windows

Je me suis rendu dans le console de recherche de **Google Chrome**, je télécharge Git et l'installer.

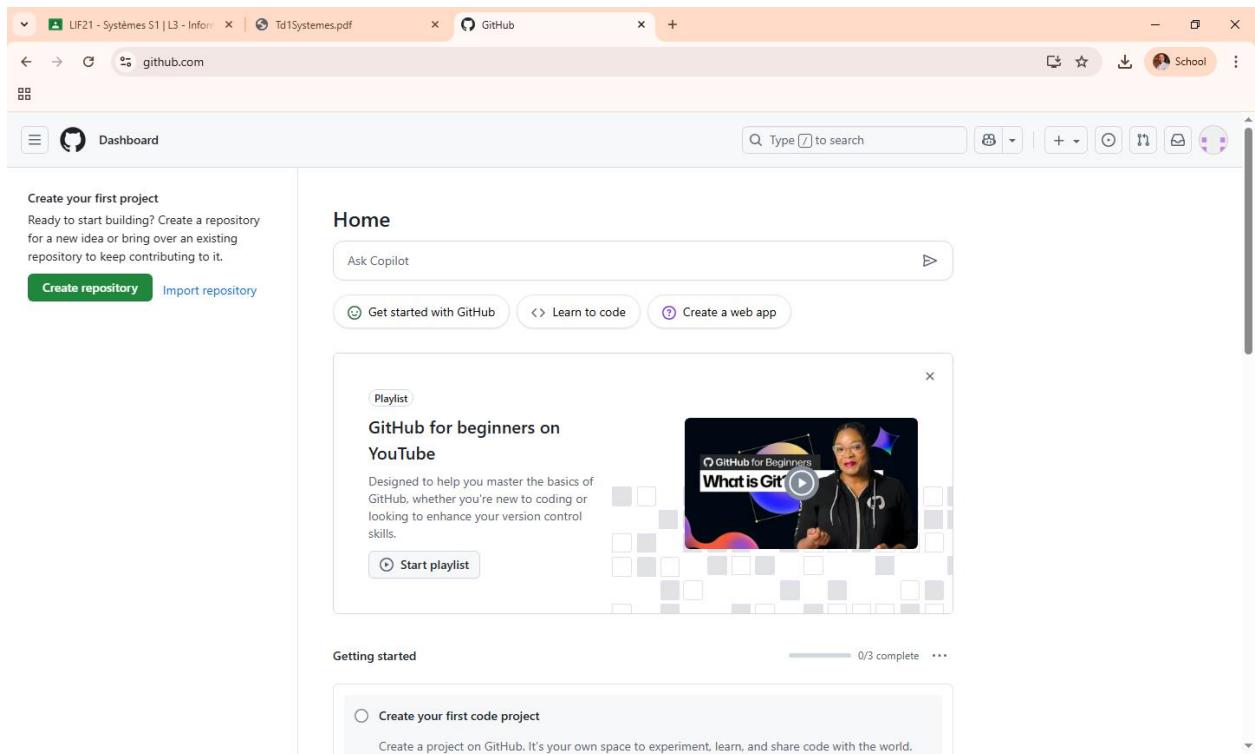
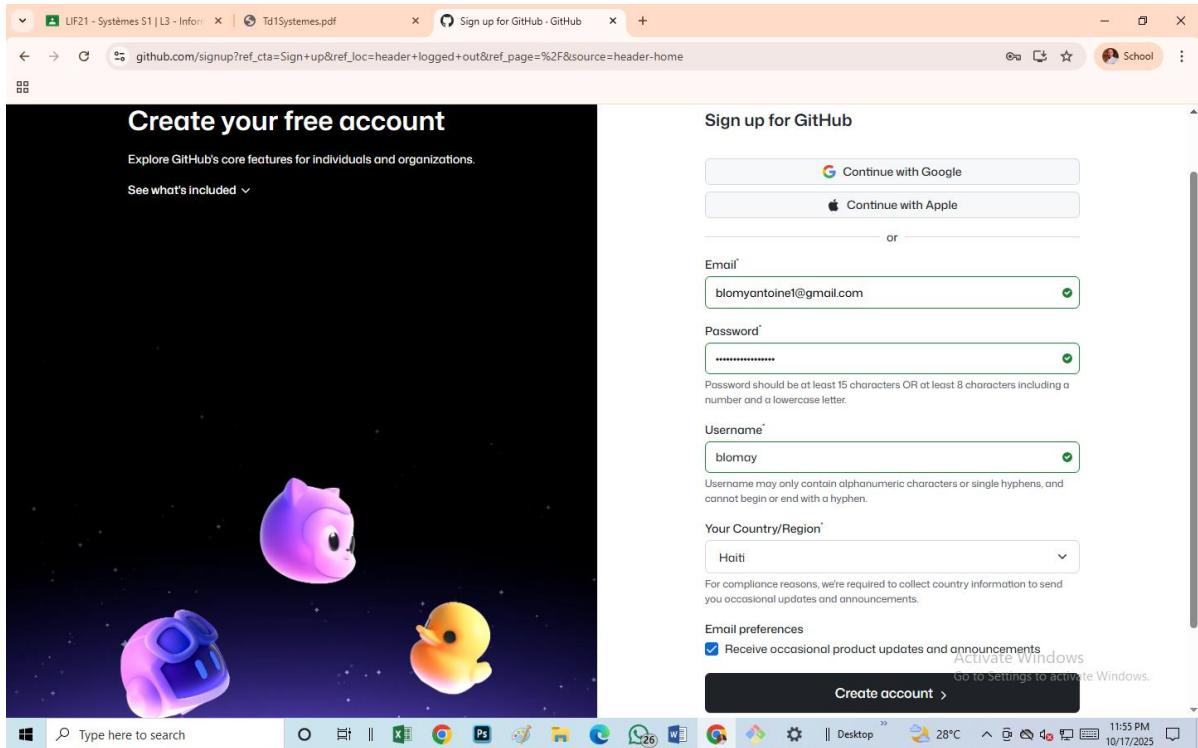


2- Vérification de l'installation de Git sur PowerShall ou sur Git



Pour le faire, nous allons dans le Terminal en tapant la commande « git - -version » puis la touche entrée. Cette commande m'a permis de vérifier que Git est bien installé sur mon ordinateur

3- Création d'un compte GitHub



Je remplis tous les champs, puis clique sur « Create account » pour créer mon compte sur GitHub.

4- Configuration de Git

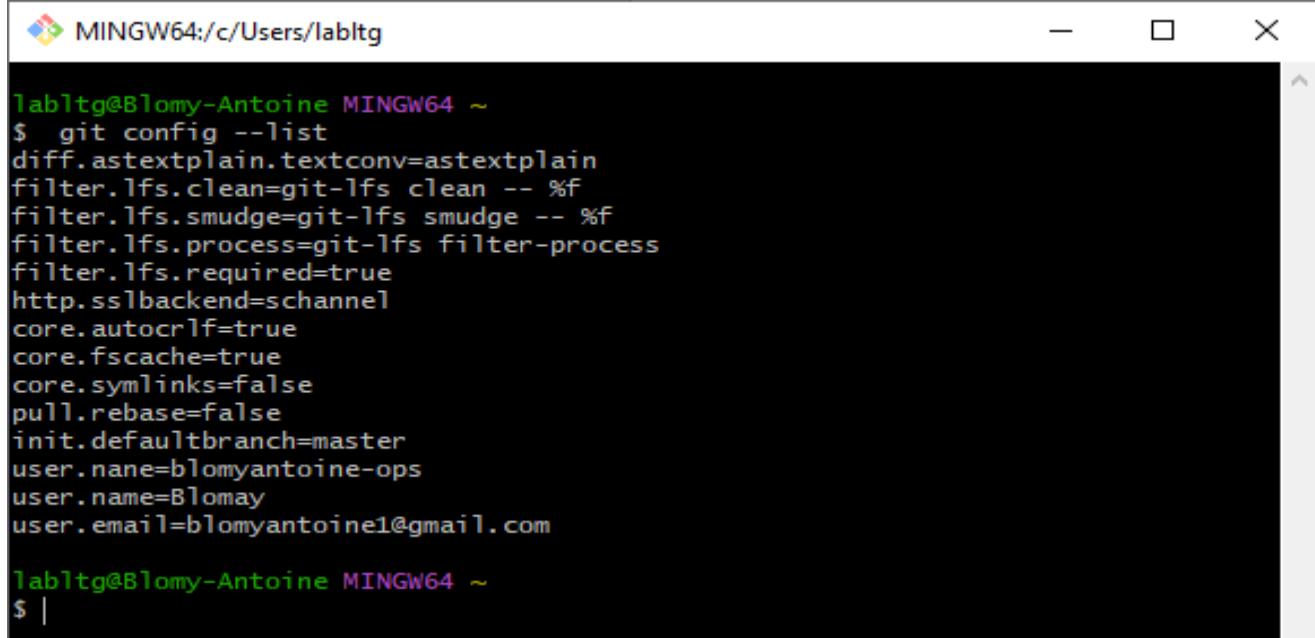
```
Tabltg@Blomy-Antoine MINGW64 ~
$ git config --global user.email "blomyantoine1@gmail.com"

Tabltg@Blomy-Antoine MINGW64 ~
$
```

git config --global user.email "ton.email@example.com" ou git config --global user.name "TonNom"

Cette commande m'a permis de configurer le compte Git.

Vérification



```
MINGW64:/c/Users/labltg
Tabltg@Blomy-Antoine MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
init.defaultbranch=master
user.name=blomyantoine-ops
user.name=Blomay
user.email=blomyantoine1@gmail.com

Tabltg@Blomy-Antoine MINGW64 ~
$ |
```

git config --list est la commande qui produit ce résultat après la vérification , nous avons vu le nom utilisateur et l'adresse électronique utilisé .

5- Je teste Git avec PowerShell

1- Je crée un dossier de Test

```
MINGW64:/c/Users/labltg
labltg@Blomy-Antoine MINGW64 ~
$ ls
'3D Objects'/
'AppData'/
'Application Data'@
'Contacts'/
'Cookies'@
'Desktop'/
'Documents'/
'Downloads'/
'Favorites'/
'Links'/
'Local Settings'@
'MES PROJETS PYTHON'/
'Music'/
'My Documents'@
NTUSER.DAT
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.blf
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000000
1.regtrans-ms
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000000
2.regtrans-ms
NetHood@
```

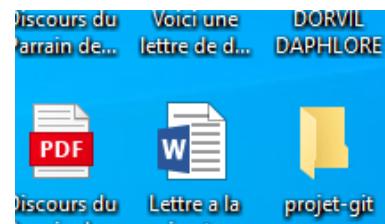
```
MINGW64:/c/Users/labltg/Desktop
2.regtrans-ms
NetHood@
OneDrive/
Pictures/
PrintHood@
Recent@
'Saved Games'/
Searches/
SendTo@
'Start Menu'@
Templates@
Videos/
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini

labltg@Blomy-Antoine MINGW64 ~
$ cd Desktop

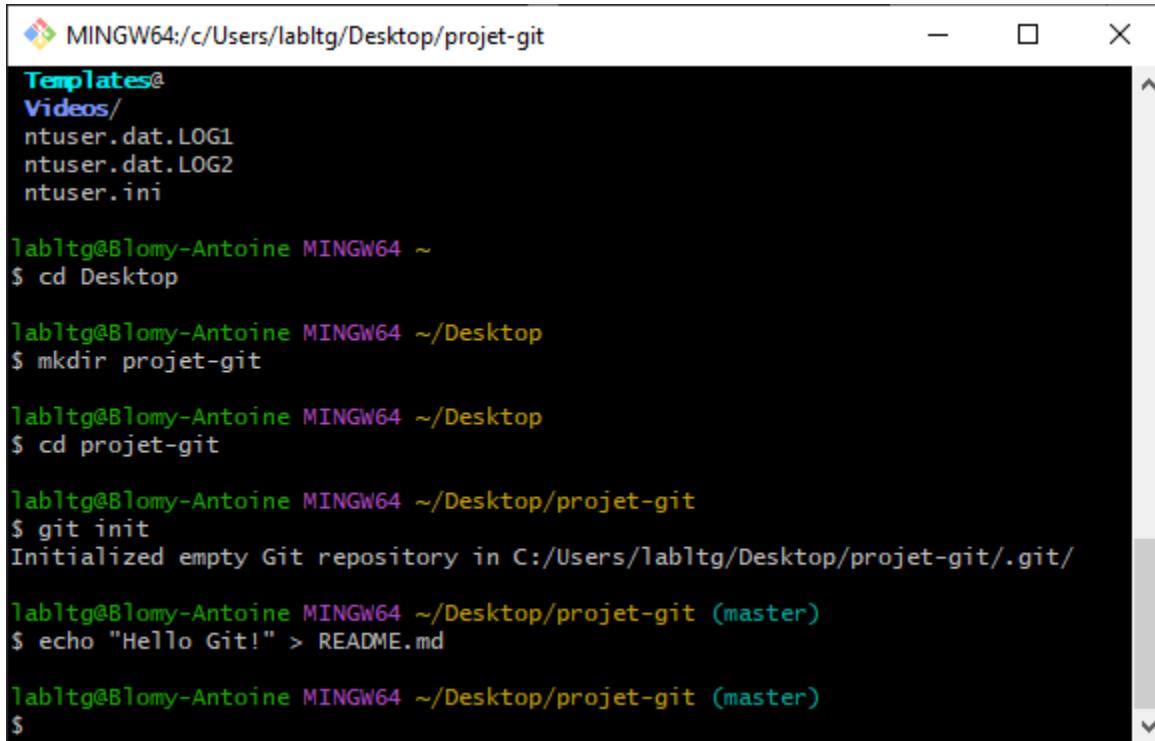
labltg@Blomy-Antoine MINGW64 ~/Desktop
$ mkdir projet-git

labltg@Blomy-Antoine MINGW64 ~/Desktop
$ |
```

Ces commandes exécutées dans le Terminal m'a permis de créer un dossier test « projet-git » sur le Desktop. Vous pouvez voir le résultat dans l'affiche suivant.



Initialisation du dépôt



```
MINGW64:/c/Users/labltg/Desktop/projet-git
Templates@ 
Videos/
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini

labltg@Blomy-Antoine MINGW64 ~
$ cd Desktop

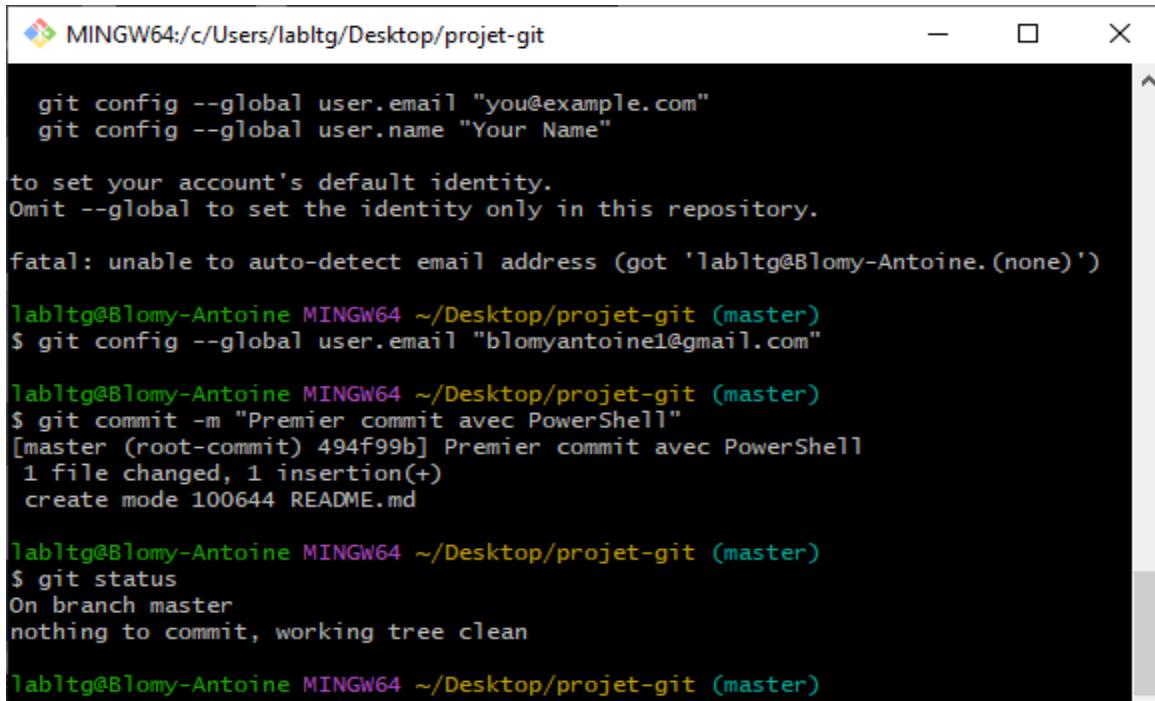
labltg@Blomy-Antoine MINGW64 ~/Desktop
$ mkdir projet-git

labltg@Blomy-Antoine MINGW64 ~/Desktop/projet-git
$ cd projet-git

labltg@Blomy-Antoine MINGW64 ~/Desktop/projet-git
$ git init
Initialized empty Git repository in C:/Users/labltg/Desktop/projet-git/.git/

labltg@Blomy-Antoine MINGW64 ~/Desktop/projet-git (master)
$ echo "Hello Git!" > README.md

labltg@Blomy-Antoine MINGW64 ~/Desktop/projet-git (master)
$
```



```
MINGW64:/c/Users/labltg/Desktop/projet-git
git config --global user.email "you@example.com"
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

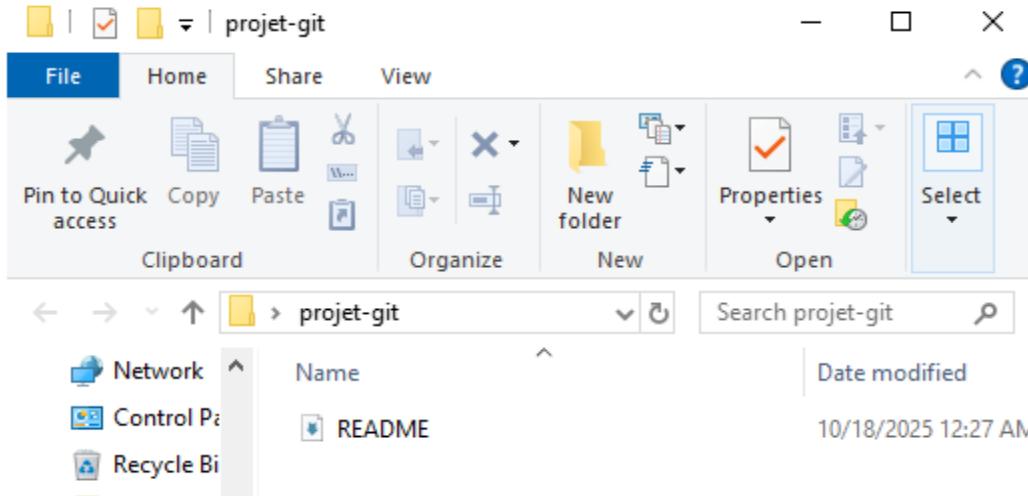
fatal: unable to auto-detect email address (got 'labltg@Blomy-Antoine.(none)')

labltg@Blomy-Antoine MINGW64 ~/Desktop/projet-git (master)
$ git config --global user.email "blomyantoine1@gmail.com"

labltg@Blomy-Antoine MINGW64 ~/Desktop/projet-git (master)
$ git commit -m "Premier commit avec PowerShell"
[master (root-commit) 494f99b] Premier commit avec PowerShell
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

labltg@Blomy-Antoine MINGW64 ~/Desktop/projet-git (master)
$ git status
On branch master
nothing to commit, working tree clean

labltg@Blomy-Antoine MINGW64 ~/Desktop/projet-git (master)
```



```
echo "Hello Git!" > README.md
git add README.md
git commit -m "Premier commit avec PowerShell"
git status
```

Ces commandes m'ont permis de faire le dépôt local dans mon dossier « projet-git ».

Historique des commit

```
labiltg@Blomy-Antoine MINGW64 ~/Desktop/projet-git (master)
$ git log
commit 494f99bf0360ba39d8a0bc1dc522307e3ccb1cc7 (HEAD -> master)
Author: Blomay <blomyantoine1@gmail.com>
Date:   Sat Oct 18 00:36:35 2025 -0400

    Premier commit avec PowerShell

labiltg@Blomy-Antoine MINGW64 ~/Desktop/projet-git (master)
$ |
```

git log est commande qui me permet de consulter l'historicité des commits.

6- Generation d'une nouvelle clé SSH

```

Tabltg@Blomy-Antoine MINGW64 ~
$ ssh-keygen -t rsa -b 4096 -C "blomyantoine1@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/labltg/.ssh/id_rsa):
/c/Users/labltg/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase for "/c/Users/labltg/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/labltg/.ssh/id_rsa
Your public key has been saved in /c/Users/labltg/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vhZoozLiNLa7Qmf6DBf5cg+FqnshafPhcL5FArMuP1E blomyantoine1@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
+---[SHA256]----+

```

`ssh-keygen -t rsa -b 4096 -C votre.email@example.com` c'est cette commande qui affiche les deux clés. Mais nous allons afficher la clé publique.

Afficher la clé publique générée

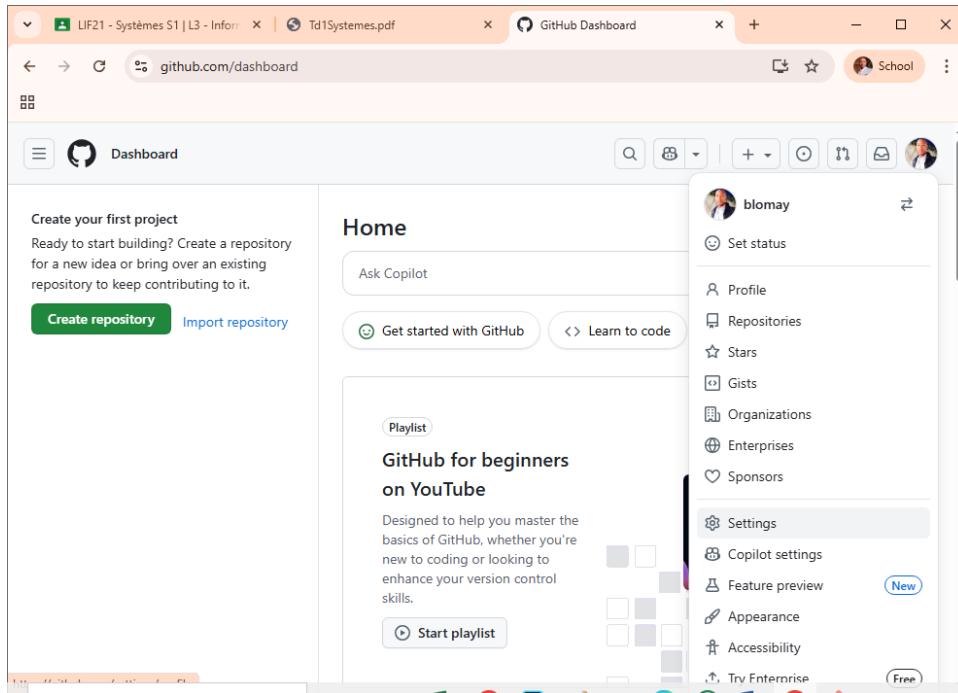
```

Tabltg@Blomy-Antoine MINGW64 ~
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDZjEU7CIr/W6CzRUjR7JjKdwKR5WeG5BMobP6ARj4H
c/7HNJYUa/sILAjR2Rb1Su0y2PS3KaTXoYDDjKTuV+mb080vQtk8vxCUIZe4w54EC3MbCoex0h1BmjBs
e903ou0eAzM4/M1n9XPJ1fPFjQoG2bweBh7X56Z7ovj1E3TiFhRa9h+yREDEGrb0J106gvwZ797yVg8d
ihgdwnNn8QG7FsTv+ZeRI0oeiadCITg+nzLXBzrQwqm0/+R+2AkXHUwONZNy58rC/0C13w3Tmiye01K
MoJVkGRUVdN22rrXw6FYWJ7MJUY6Ya9u50Me48962voWiuzLoWcM+jue+VzuIvCFBtkbyvG0PigYsoL
Haprwb5M1MKjccZvGs9UUtmVACbF1iNlPiMEsiUbyBFuSwazY/++/VIwW8/AoFAVJo+Xo9C5AZ7v7qNGp
1rSgNDH+26HrS79KwrtxsI6tp/gYYb5NT5GjUdW6nhAhapmY4SkalWGFtTd6ZUz++1oUoILGwVCjm9FEh
IOXKz8XwLSMAAENEeqKp+jCnmbC/J4xc9MI9zqvDz9rYCkUy7WNnZf2L3byOSeFzWrRNk3UATcaPYAY6
S1Qsh0kLU+w4s5CxWbHqsMX5qwDL6Lf0I0ayYjFUHeLhNhvVfY1gqb4r7k2vVBvwfRUSaRSkEMWPJcsZ
CQ== blomyantoine1@gmail.com

Tabltg@Blomy-Antoine MINGW64 ~
$ |

```

`cat ~/.ssh/id_rsa.pub` est commande qui donne accès à la clé SSH publique . C'est cette clé que nous devons rendre publique.



The screenshot shows the 'Your profile' settings page for the user 'blomay'. The 'Public profile' tab is selected. It displays the user's name, profile picture, public email (which is a placeholder for a verified email), bio (a text input field), and pronouns (set to 'Don't specify'). On the left, there is a sidebar with sections for Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and licensing, Emails, Password and authentication, Sessions, SSH and GPG keys, Organizations, Enterprises, and Moderation.

Cette étape nous a permis de mettre la SSH publique sur notre compte GitHub. j'ai ensuite copié la clé publique dans GitHub, section Settings : « Setting » puis « SSH and GPG keys ».

You have successfully added the key 'key win'.

blomay (blomay)
Your personal account

[Go to your personal profile](#)

[New SSH key](#)

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

key win	SHA256:QxEixpXIPZ60e++VLmoigC/rOfkZVogivgEk/ShGf+c	Delete
SSH	Added on Oct 18, 2025	
	Never used — Read/write	

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

[New GPG key](#)

GPG keys

Voici le résultat final. La clé SSH publique est activée. Mon compte est authentifié.

7- Crédation d'un dépôt GitHub

You have successfully added the key 'key win'.

blomay (blomay)
Your personal account

[New issue](#)

[New repository](#)

[Import repository](#)

[New codespace](#)

[New gist](#)

[New organization](#)

[New project](#)

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

key win	SHA256:QxEixpXIPZ60e++VLmoigC/rOfkZVogivgEk/ShGf+c	Delete
SSH	Added on Oct 18, 2025	
	Never used — Read/write	

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

[New GPG key](#)

GPG keys

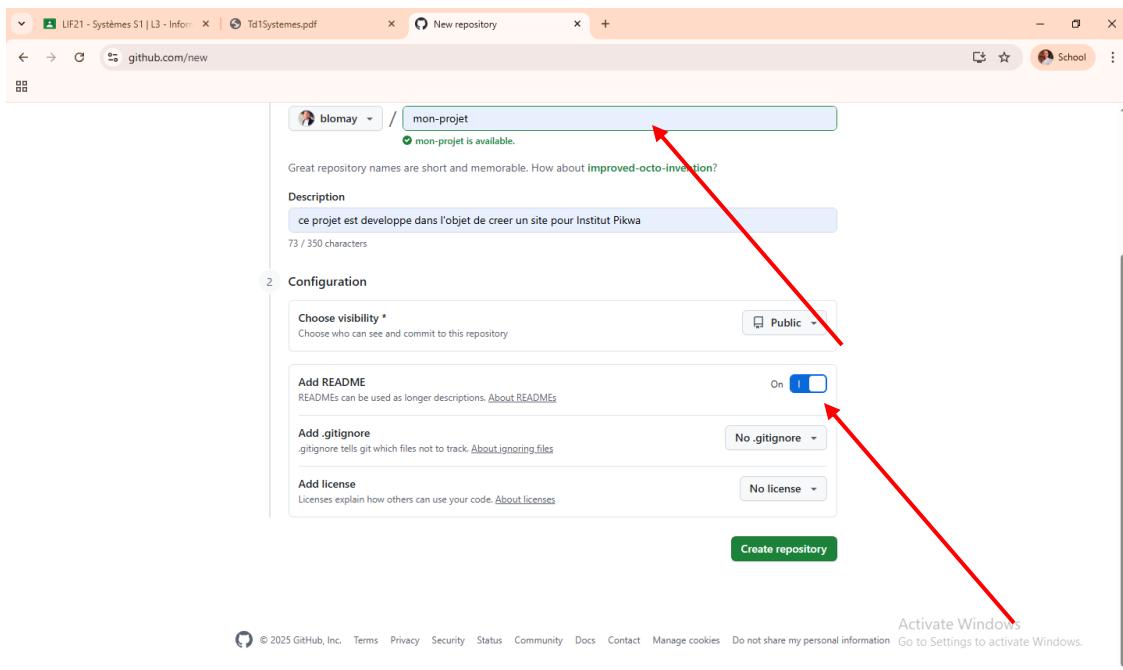
There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

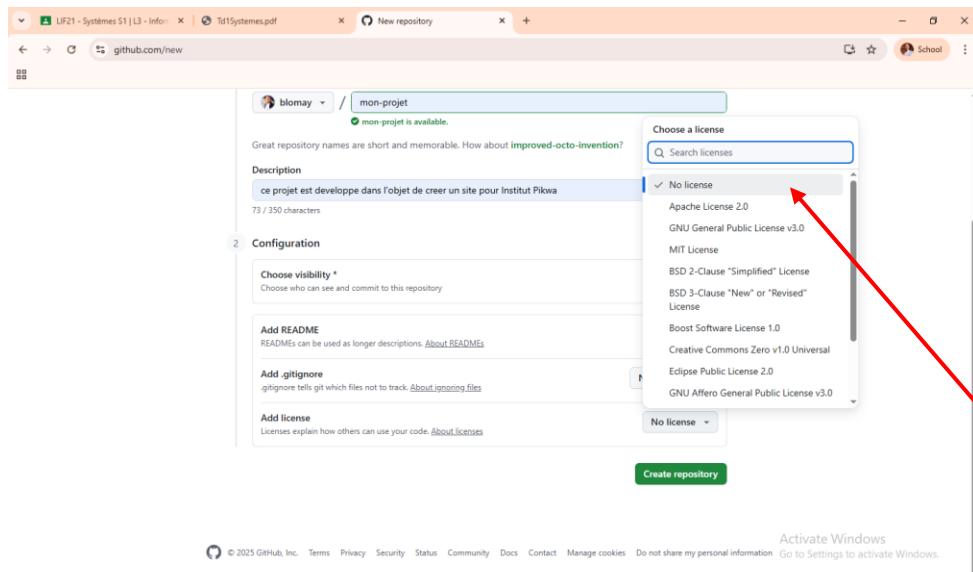
[Activate Windows](#)
Go to Settings to activate Windows.

Vigilant mode

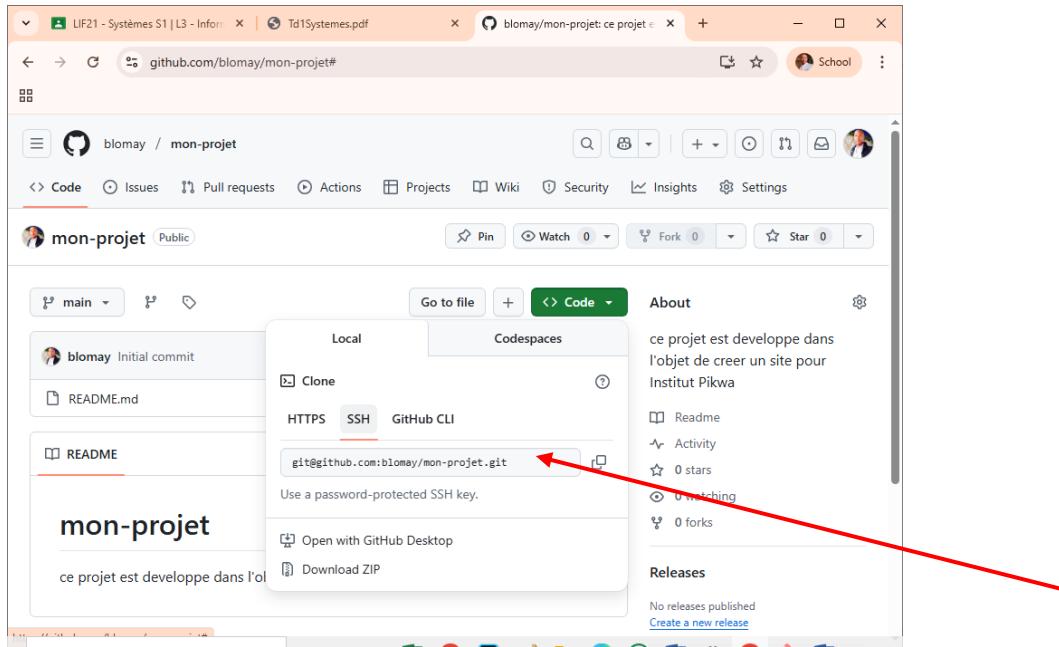
On click sur le signe +, puis New repository



J'ai ajouté un
fichier
README.md



8- Cloner un dépôt GitHub via SSH

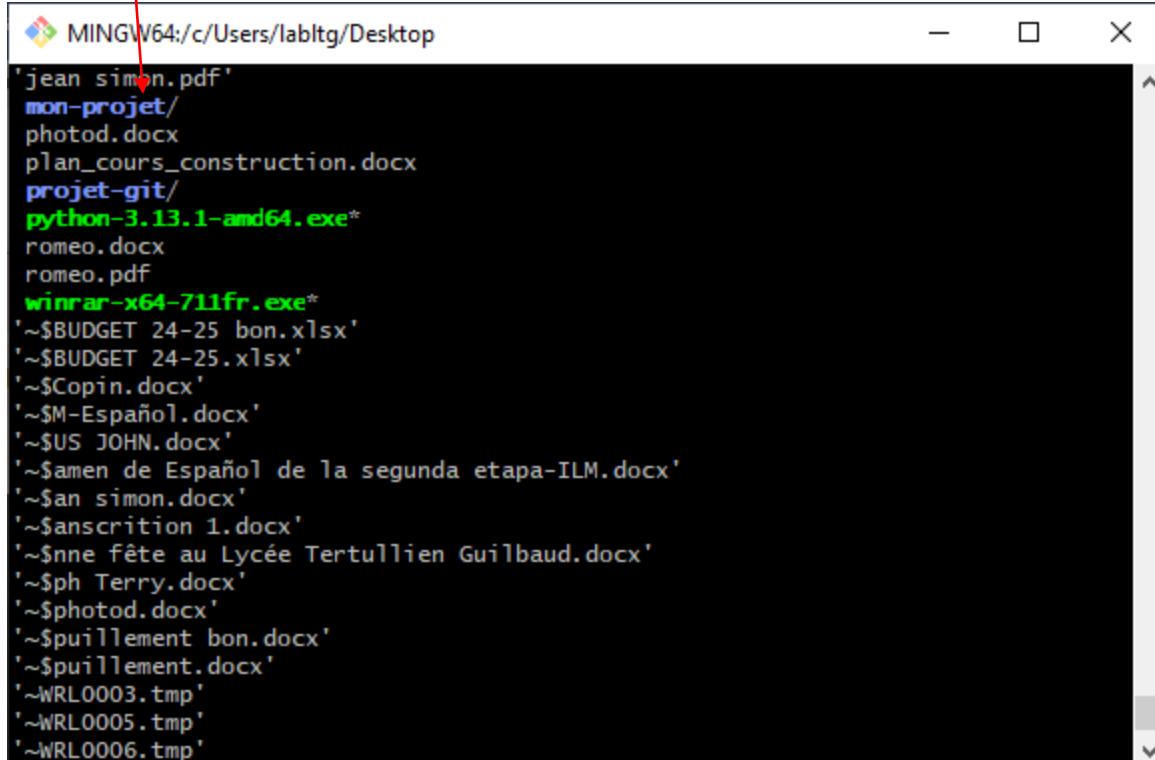


Je recopie le lien SSH pour aller cloner mon dossier : « **mon-projet** ».

```
Tabltg@Blomy-Antoine MINGW64 ~/Desktop
$ git clone git@github.com:Blomay/mon-projet.git
Cloning into 'mon-projet'...
The authenticity of host 'github.com (140.82.113.4)' can't be established.
ED25519 key fingerprint is: SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

Tabltg@Blomy-Antoine MINGW64 ~/Desktop
$
```

git clon git@github.com:username/mon-projet.git cette démarche me permet de cloner le dossier « **mon-projet** » avec la clé SSH.



```

MINGW64:/c/Users/labltg/Desktop
'jean simon.pdf'
'mon-projet/'
photod.docx
plan_cours_construction.docx
projet-git/
python-3.13.1-amd64.exe*
romeo.docx
romeo.pdf
winrar-x64-711fr.exe*
'~$BUDGET 24-25 bon.xlsx'
'~$BUDGET 24-25.xlsx'
'~$Copin.docx'
'~$M-Español.docx'
'~$US JOHN.docx'
'~$amen de Español de la segunda etapa-ILM.docx'
'~$an simon.docx'
'~$anscrition 1.docx'
'~$nne fête au Lycée Tertullien Guilbaud.docx'
'~$ph Terry.docx'
'~$photod.docx'
'~$puissement bon.docx'
'~$puissement.docx'
'~$WRL0003.tmp'
'~$WRL0005.tmp'
'~$WRL0006.tmp'

```

```

labltg@Blomy-Antoine MINGW64 ~/Desktop
$ cd mon-projet

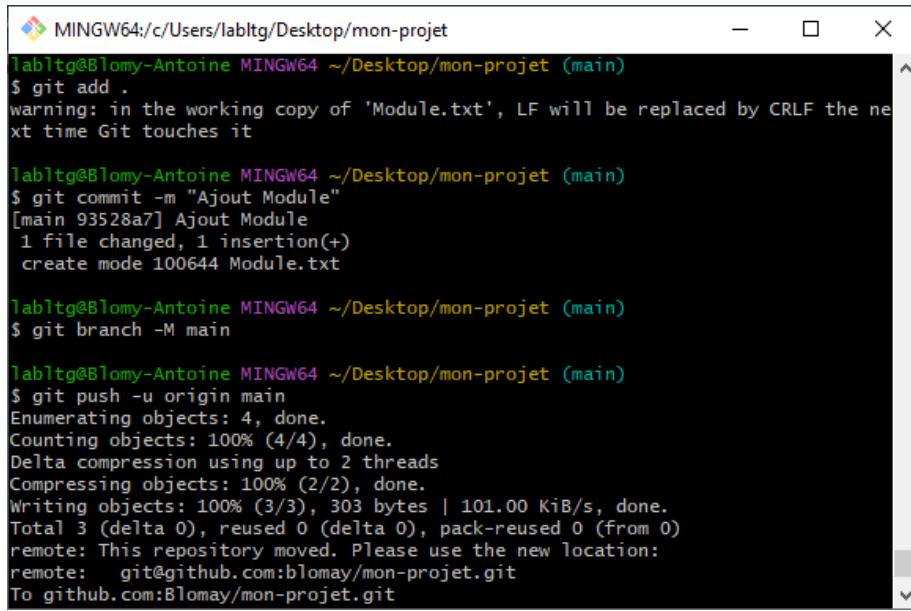
labltg@Blomy-Antoine MINGW64 ~/Desktop/mon-projet (main)
$ echo "Système d'exploitation!" > Module.txt

labltg@Blomy-Antoine MINGW64 ~/Desktop/mon-projet (main)
$ git add .
warning: in the working copy of 'Module.txt', LF will be replaced by CRLF the next time Git touches it

labltg@Blomy-Antoine MINGW64 ~/Desktop/mon-projet (main)
$ |

```

J'ai ajouté et envoyé des fichiers vers GitHub.



```

MINGW64:/c/Users/labltg/Desktop/mon-projet
labltg@Blomy-Antoine MINGW64 ~/Desktop/mon-projet (main)
$ git add .
warning: in the working copy of 'Module.txt', LF will be replaced by CRLF the next time Git touches it

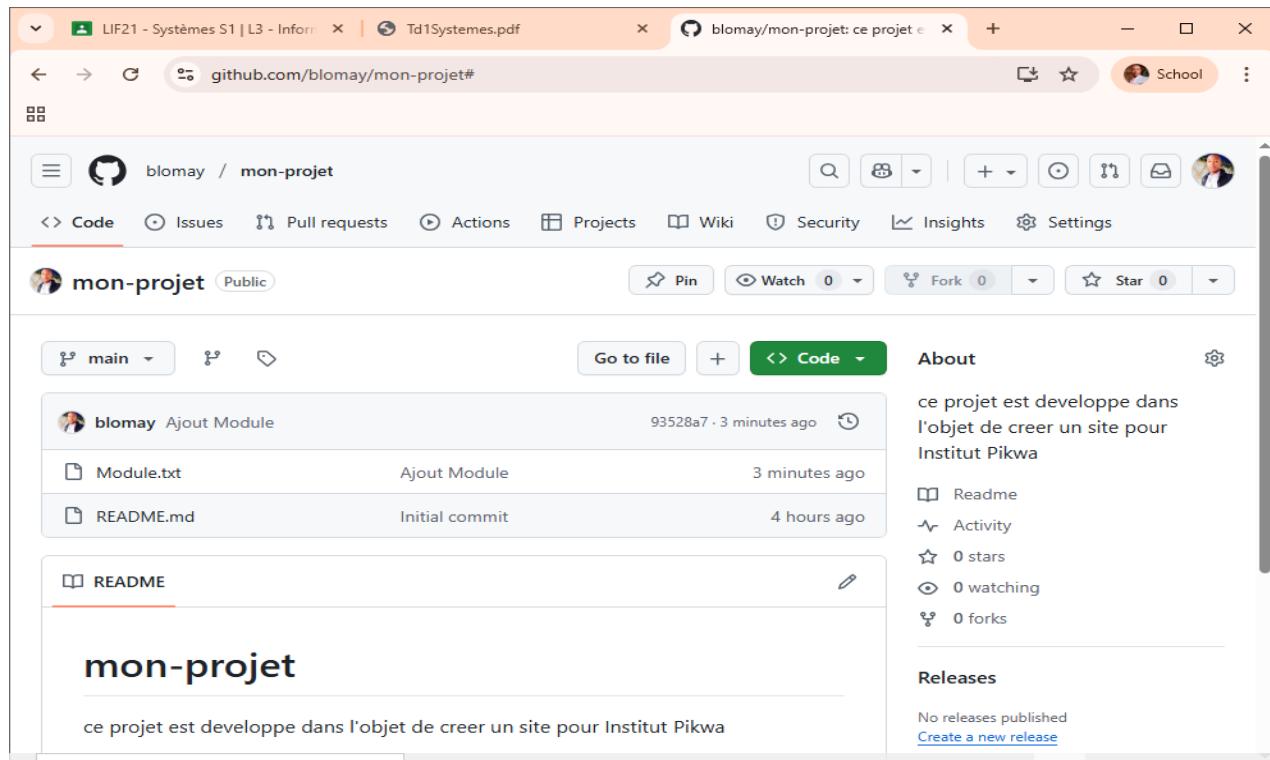
labltg@Blomy-Antoine MINGW64 ~/Desktop/mon-projet (main)
$ git commit -m "Ajout Module"
[main 93528a7] Ajout Module
 1 file changed, 1 insertion(+)
 create mode 100644 Module.txt

labltg@Blomy-Antoine MINGW64 ~/Desktop/mon-projet (main)
$ git branch -M main

labltg@Blomy-Antoine MINGW64 ~/Desktop/mon-projet (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 101.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: This repository moved. Please use the new location:
remote:   git@github.com:blomay/mon-projet.git
To github.com:Blomay/mon-projet.git

```

J'ai ajouté et envoyé des fichiers vers GitHub.



Résultats : j'ai vérifié que les fichiers ont été ajouté.

```
cd Desktop  
git clone git@github.com:username/mon-projet.git  
ls  
cd mon-projet  
echo "Système d'exploitation!" > Module.txt  
git add .  
git status  
git commit -m "Ajout Module"  
git branch -M main  
git push -u origin main
```

Ces démarches m'ont permis de Cloner un dépôt GitHub via SSH. Tel le dépôt mon-projet.

Conclusion

A la fin ce travail Dirigé, nous avons appris à :

- Installer et configurer Git sur Windows.
- Créer un dépôt local et effectuer des commits
- Générer et utiliser un clé SSH pour authentifier mon compte GitHub
- Créer, cloner et synchroniser un dépôt distant GitHub.
- Organiser un projet avec une structure claire et héberger mon rapport en ligne.

Comme difficultés rencontrées, la génération de la clé SSH à GitHub ont nécessité plusieurs essais. Quelques erreurs de syntaxes dans les commandes ont été corrigées grâce à la documentation Git.

Nous pouvons dire que toutes les étapes ont été reproduites avec succès, et le rapport final a été hébergé sur GitHub.