



IUS
INSTITUT
UNIVERSITAIRE
DES SCIENCES

FALCUTÉ DES SCIENCES ET TECHNOLOGIES (FST)

TROISIÈME ANNÉE

Rapport du travail de Laboratoire N° 1

Cours : Systèmes

Étudiante : Christy Gérys LAMBERT

Professeur : Ismaël SAINT AMOUR

LE 19 OCTOBRE 2025

L'objectif de ce TD est de :

- Comprendre les concepts de base de Git (système de gestion de versions).
- Savoir utiliser Git pour gérer un projet localement.
- Découvrir GitHub et apprendre à collaborer sur un dépôt distant

Travail Dirigé

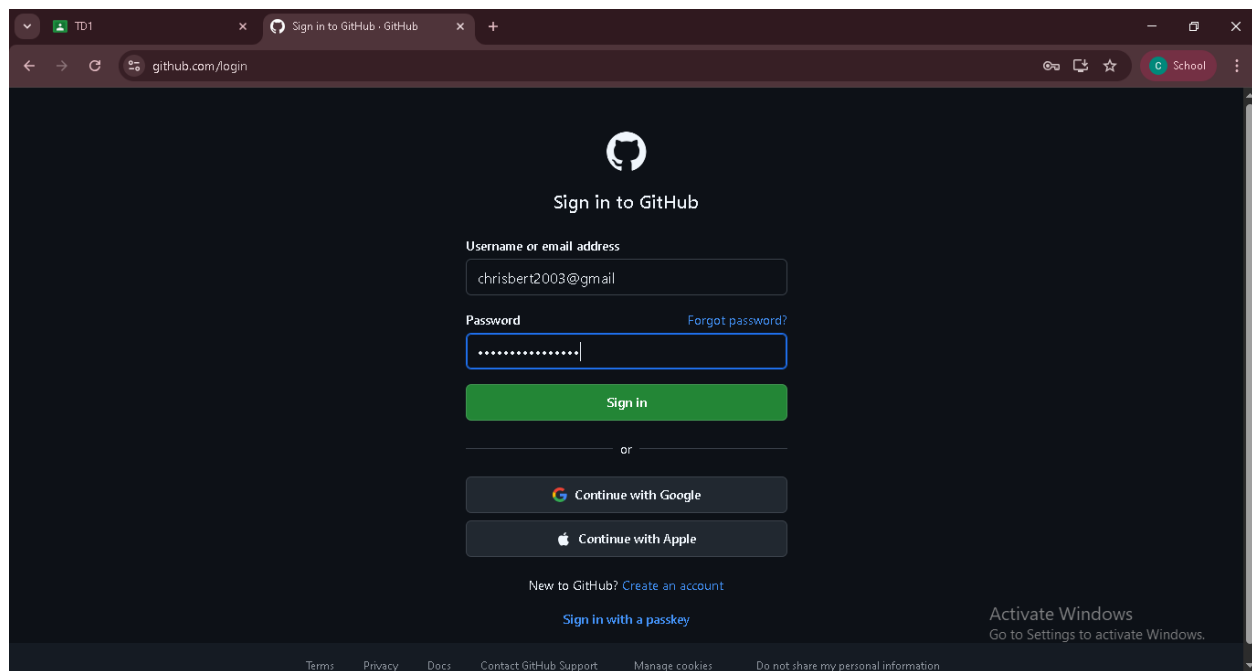
- Reproduire une série de tâches.
- Créer un nouveau dépôt sur GitHub.

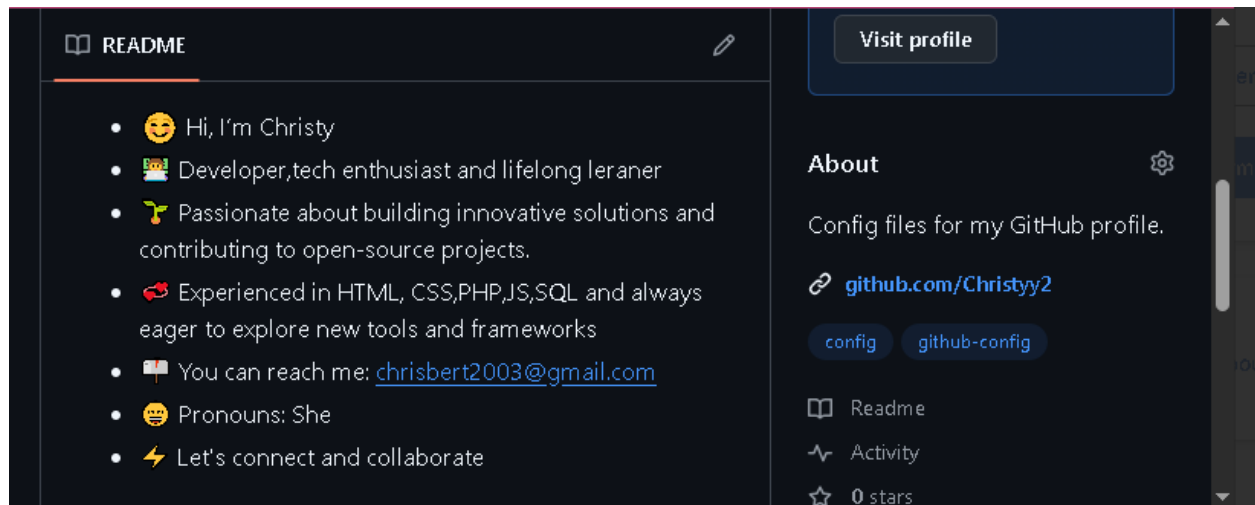
1) Vérification de la version Git installée sur le PC.



```
PC@DESKTOP-U5501MB MINGW32 ~  
$ git --version  
git version 2.46.1.windows.1  
  
PC@DESKTOP-U5501MB MINGW32 ~  
$
```

2) Deuxième étape : Connexion à mon compte GitHub





3) Troisième étape : Configuration de Git

```
MINGW32:/c/Users/PC/Desktop/projet1-git

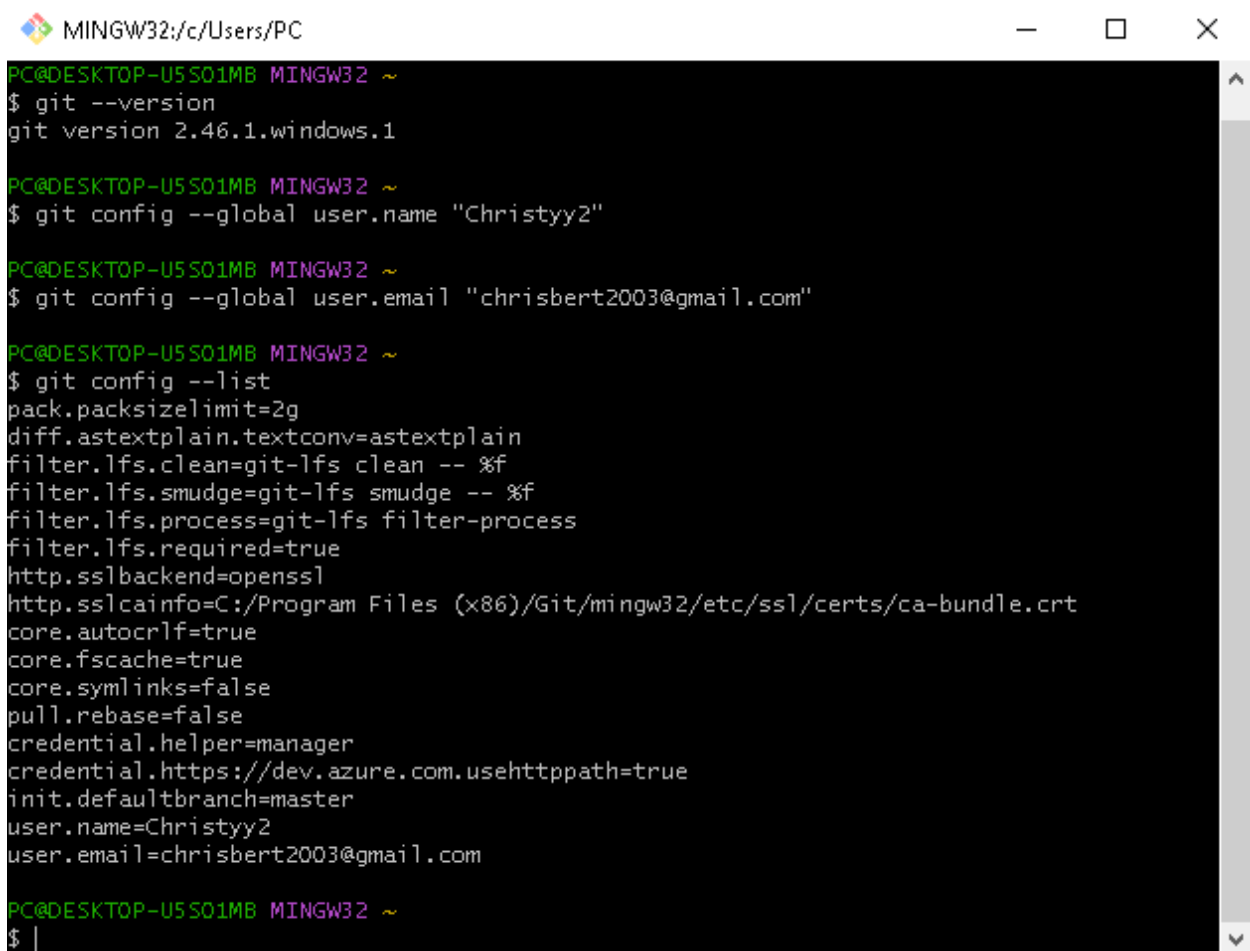
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ git --version
git version 2.46.1.windows.1

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ git config --global user.name "Christyy2"

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ git config --global user.email "chrisbert2003@gmail.com"

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ |
```

4) Quatrième étape : Vérification des données utilisateurs



```
MINGW32:/c/Users/PC
PC@DESKTOP-U5S01MB MINGW32 ~
$ git --version
git version 2.46.1.windows.1

PC@DESKTOP-U5S01MB MINGW32 ~
$ git config --global user.name "Christyy2"

PC@DESKTOP-U5S01MB MINGW32 ~
$ git config --global user.email "chrisbert2003@gmail.com"

PC@DESKTOP-U5S01MB MINGW32 ~
$ git config --list
pack.packsizelimit=2g
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=openssl
http.sslcainfo=C:/Program Files (x86)/Git/mingw32/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Christyy2
user.email=chrisbert2003@gmail.com

PC@DESKTOP-U5S01MB MINGW32 ~
$ |
```

5) Cinquième étape : Affichage de la liste des fichiers et dossiers avec la commande ls

```
MINGW32:/c/Users/PC
PC@DESKTOP-U5S01MB MINGW32 ~
$ ls
'3D Objects'/
AppData/
'Application Data'@
Contacts/
Cookies@
Desktop/
DevConnect-ameliore.epgz
DevConnect.epgz
Documents/
Downloads/
Favorites/
IntelGraphicsProfiles/
Links/
'Local Settings'@
Music/
'My Documents'@
NTUSER.DAT
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.b1f
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000001.regtr
ans-ms
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000002.regtr
ans-ms
NetHood@
OneDrive/
Pictures/
PrintHood@
PyCharmMiscProject/
PycharmProjects/
QuizFlashApp.epgz
```

```
MINGW32:/c/Users/PC/Desktop
OneDrive/
Pictures/
PrintHood@
PyCharmMiscProject/
PycharmProjects/
QuizFlashApp.epgz
Recent@
Resources/
'Saved Games'/
Searches/
SendTo@
'Start Menu'@
Templates@
Videos/
activite1/
developementMobile/
domino_files.txt
donnees.txt
flutter
index.html
jcef_8252.log
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
pages/
projet1_flutter/
source/
test.c
test1.c
PC@DESKTOP-U5S01MB MINGW32 ~
$ cd Desktop
```

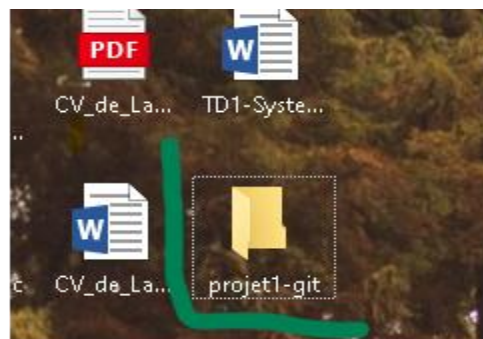
6) Sixième étape : Création d'un dossier dans Desktop

```
MINGW32:/c:/Users/PC/Desktop
Recent@
Resources/
'Saved Games'/
Searches/
SendTo@
'Start Menu'@
Templates@
Videos/
activitel/
developementMobile/
domino_files.txt
donnees.txt
flutter
index.html
jcef_8252.log
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
pages/
projet1_flutter/
source/
test.c
test1.c

PC@DESKTOP-U5S01MB MINGW32 ~
$ cd Desktop

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ mkdir projet1-git

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$
```



7) Septième étape : Entrée dans notre dossier « projet1-git » fraîchement créé.

```
MINGW32:/c/Users/PC/Desktop/projet1-git
Searches/
SendTo@
Start Menu*
Templates@
Videos/
activitel/
developementMobile/
domino_files.txt
donnees.txt
flutter
index.html
jcef_8252.log
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
pages/
projet1_flutter/
source/
test.c
test1.c

PC@DESKTOP-U5S01MB MINGW32 ~
$ cd Desktop

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ mkdir projet1-git

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ cd projet1-git

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git
$
```

8) Huitième étape : Initialisation du dépôt git.

```
MINGW32:/c/Users/PC/Desktop/projet1-git
Videos/
activitel/
developementMobile/
domino_files.txt
donnees.txt
flutter
index.html
jcef_8252.log
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
pages/
projet1_flutter/
source/
test.c
test1.c

PC@DESKTOP-U5S01MB MINGW32 ~
$ cd Desktop

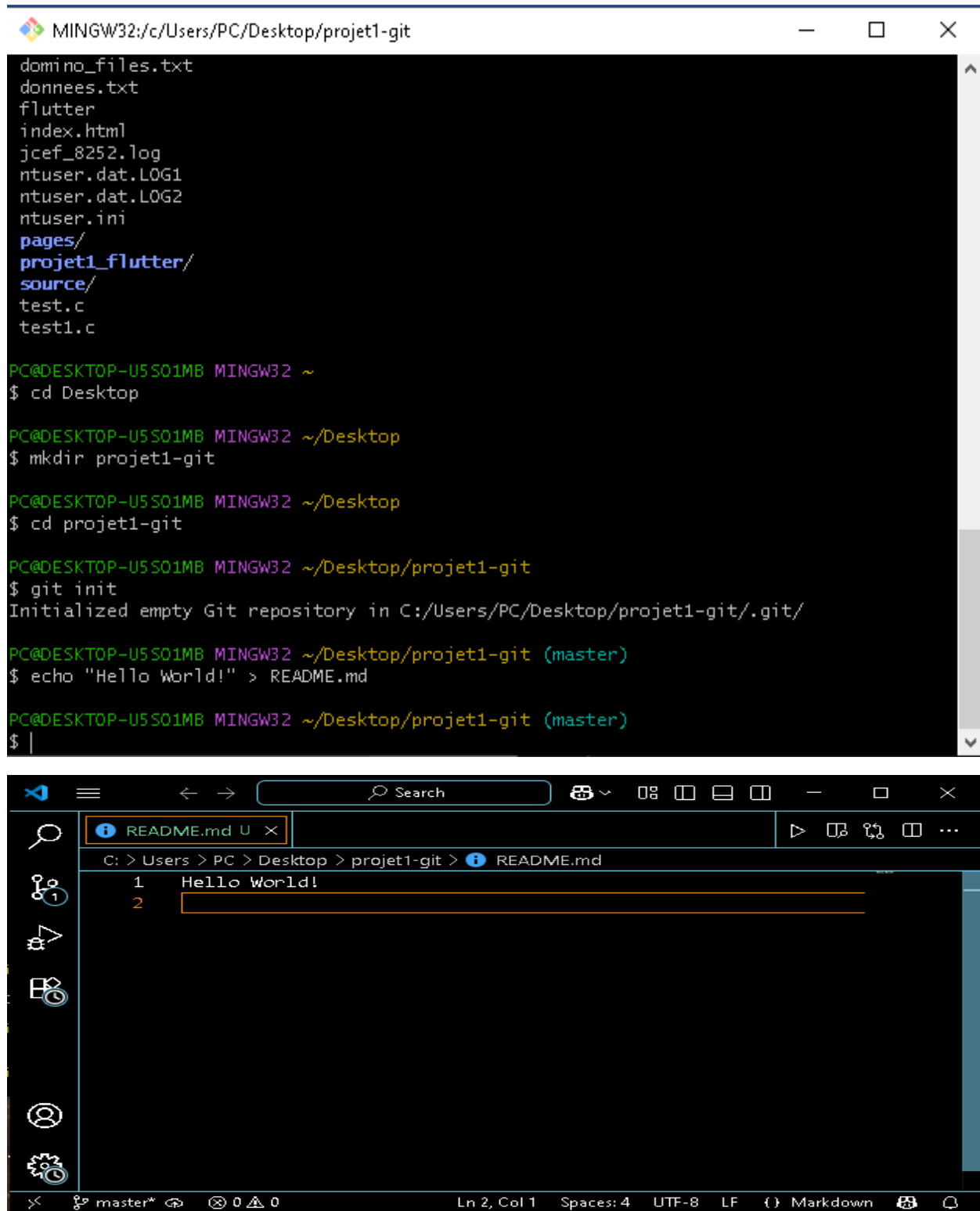
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ mkdir projet1-git

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ cd projet1-git

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git
$ git init
Initialized empty Git repository in C:/Users/PC/Desktop/projet1-git/.git/

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$
```


9) Neuvième étape : Ouverture et écriture de mon projet dans le fichier README.md



The image shows two screenshots illustrating the steps to create and edit a README.md file in a new Git project.

Terminal Window (MINGW32):

```
domino_files.txt
donnees.txt
flutter
index.html
jcef_8252.log
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
pages/
projet1_flutter/
source/
test.c
test1.c

PC@DESKTOP-U5S01MB MINGW32 ~
$ cd Desktop

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ mkdir projet1-git

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ cd projet1-git

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git
$ git init
Initialized empty Git repository in C:/Users/PC/Desktop/projet1-git/.git/

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ echo "Hello World!" > README.md

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ |
```

VS Code Editor:

The VS Code editor shows the file explorer on the left with the project structure:

- 1 README.md U x

The main editor area displays the content of README.md:

```
1 Hello World!
2
```

The status bar at the bottom indicates the current file is README.md, on the master branch, with 0 changes, 0 deletions, and 0 additions. The editor is in UTF-8 encoding with LF line endings and Markdown format.

10)Dixième étape : Ajout du dossier README.md dans Git et premier commit effectué

```
MINGW32:/c/Users/PC/Desktop/projet1-git

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ cd projet1-git

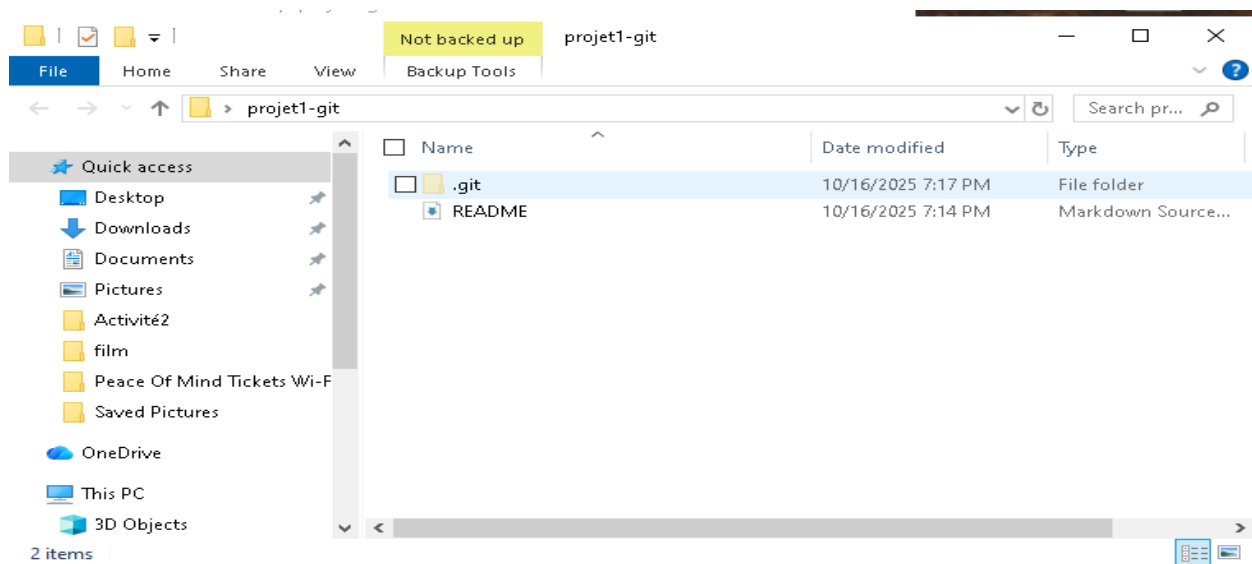
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git
$ git init
Initialized empty Git repository in C:/Users/PC/Desktop/projet1-git/.git/

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ echo "Hello World!" > README.md

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ git commit -m "Premier commit"
[master (root-commit) ab2ba86] Premier commit
1 file changed, 1 insertion(+)
create mode 100644 README.md

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$
```



11) Onzième étape : Affichage de l'historique des commits dans un dépôt Git.

```
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ git log
commit ab2ba86bf8e069bfb0e67300750e7859b835e8cd (HEAD -> master)
Author: Christyy2 <chrisbert2003@gmail.com>
Date:   Fri Oct 17 10:40:59 2025 -0400

    Premier commit

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ |
```

12) Douzième étape : Génération d'une clé SSH

```
MINGW32:/c/Users/PC/Desktop/projet1-git
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ ssh-keygen -t rsa -b 4096 -C "chrisbert2003@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/PC/.ssh/id_rsa):
/c/Users/PC/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/PC/.ssh/id_rsa
Your public key has been saved in /c/Users/PC/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:RZUSJmmvesk35pq+GwWTyiT2lYdoE3kScXaeobLQzDU chrisbert2003@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
|      ++E.*o..      |
|      +O*+%.O.      |
|      + X+O.=.      |
|      . B * =.      |
|      = S..         |
|      ..           |
|      O..          |
|      . +O+        |
|      o*B..         |
+-----[SHA256]-----+
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$
```

13) Treizième étape : Affichage de la clé privée avec la commande cat

```
MINGW32:/c:/Users/PC/Desktop/projet1-git
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$ cat ~/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktbjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAadzczgtcn
NhAAAAAwEAAQAAAEAE5vB8tXMGpQpd/2Wgc4pP7N2ZXFRdaa5szad71GKxCyHMsH5BANE
oxqxz4Z0+8muRkz321oXMDHgmqfCVLbDBGfzG4wNEMELyAV19BKyvGEDdD0ZWB62y7zPH
kYj9kweIFgAWB/+n2F5FF2CCHuN1mwo0cIVTXM2jkwfXw1YgWODFoZckVixVIJTCw4toCs
CdDgCkIBhCU7pzHb9vNohjKG1x45XxFYsvfKMCXLQTPGngA2oD5HPrDwKNWPWbKGMuPORJ
z3ZxHTs5t07J0Kw35p2BxZMX1eZZnARL0KpjQ85CkzmAw2miu5Cu1irqb6n6/FltZTsSka
z735t1XnJ+JDd4nyAxH7BGbJ195q5GjdJv4z1aTYjUkL0/N7JY+GfcgdwmSI5je8EbV1gm
YSN4GdfZJFrg7/6MLq07SRM9BbtGTh0nDcmESAg3MLgTUEwWwKrH+EA0dNkEggjm2hbrP6
Nirjp0q+1SduB8nkDJ60P/WTWzhzIuv4iSN11GCBuPU8F7h8mPdF/dw2RfQ6MPGzFXW
9mGoJ3oRP+ESgRGX8zt0Yyn/z3xxiMIOxhRbiLGLYqK1R9oH4u6t0WrVM24ZyErWmQs4+
B3r3tSm+3A6wSfrk1Xep7cWYBuYn/ciBikOG2YqibjC9vat8qCxi+wtK8XA3dq5EL4tuS
MAAADQUXQ5u1F00boAAAAHc3NoLXJzYQAAAEAE5vB8tXMGpQpd/2Wgc4pP7N2ZXFRdaa5s
zad71GKxCyHMsH5BANEsoxqxz4Z0+8muRkz321oXMDHgmqfCVLbDBGfzG4wNEMELyAV19
BKyvGEDdD0ZWB62y7zPHkYj9kweIFgAWB/+n2F5FF2CCHuN1mwo0cIVTXM2jkwfXw1YgW
ODFoZckVixVIJTCw4toCsCdDgCkIBhCU7pzHb9vNohjKG1x45XxFYsvfKMCXLQTPGngA2o
D5HPrDwKNWPWbKGMuPORJz3ZxHTs5t07J0Kw35p2BxZMX1eZZnARL0KpjQ85CkzmAw2miu
5Cu1irqb6n6/FltZTsSkaz735t1XnJ+JDd4nyAxH7BGbJ195q5GjdJv4z1aTYjUkL0/N7J
Y+GfcgdwmSI5je8EbV1gmYSN4GdfZJFrg7/6MLq07SRM9BbtGTh0nDcmESAg3MLgTUEwW
KrH+EA0dNkEggjm2hbrP6Nirjp0q+1SduB8nkDJ60P/WTWzhzIuv4iSN11GCBuPU8F7h
8mPdF/dw2RfQ6MPGzFXW9mGoJ3oRP+ESgRGX8zt0Yyn/z3xxiMIOxhRbiLGLYqK1R9oH4
u6t0WrVM24ZyErWmQs4+B3r3tSm+3A6wSfrk1Xep7cWYBuYn/ciBikOG2YqibjC9vat8q
Cxi+wtK8XA3dq5EL4tuSMAAADAUABAAACACop/OLru0taHLZcFXDz7YrMHMvMKV/5sJDM
Iu4whK2KUpIowvrN1Q6FebBvj6018fUw6VHLAA/ej0iMX/QGvir6agjcl117X0CzW194K
Hz/Y2+0FjoCBZLmKpSW7srxs9Bo8MsmME1uMpr3RZ5NOIiFv+AoRNZqsiVb1zp1TNz2y8D
JkE4TOPE6iDzp+kffdl71QzLKrZw1RX5Jb0cyeJQwyFPq10U/j73LRzugvzNWB0noScSk
q2aGq0Ly0x0AmrkQXy+LKvTQ0uTncgKwYtZurHpjtEbF8z1qhWYMN544qvZpFJR6Z1MK
G2pS39Bwch348z10dl75TulYB0RvYr0Z6M6cfKU0F5uJefeb/4GpYdfnczAqWUz35OHT3f
a0YBDkhY1xZqcLDMemeemDwtj0X7dj4f50LjZWtXgh7TZAkgtGzaIw9E8GEOCqMdtwFB+
+8A+qN/jCyrzB6w95joiijvp4bx1Bcbg6z38u1BxHL8E9khSpXIZh1P4spVPMjMB8cmb3
3mXV59sYRjwdw81jgBAAABAQCyZZoVv3nTLgy91WznR08fVTdIdvFFkz0CaMEFPFR1BqKc
+yfBNCCgbAQ6V0K9FAHfzHVwxZDfQuQIfTo10Ebox9k/OCn01+TfgTw/K6gkBndQhji0Q+
R2ixvvzq3yHF1Myx4VBykhs0wNa2r5ampJUUEntGKdef7DGj4X6xSLAPjtX0oBk/sU4TqI
W1/NTJONvoCaaKR7z4eM0twmwSLfKH8y+jf+buJlr/gSFFSJCazVDrTo6gPUmUvso4vuIO
8qQDONP2LMj1yQobD83z78KM5ZeqmP0eurs3YsCTL/b0E0QmcMbYbTat2f8xHSTV8QYIHK
JcrhMmTsvAFwXh4nAAABAQD5SL1nfPaZyMuYaK6j208HEbyBnwzbnjV2kYFwmbe3cG0KXb
cmiVX4+A4RFFtgJzvv03LQ1utzSFWJ0/Nuk7Q9zNc+d8S5jtm1XFpJXk6ysjNEzSpzJStS
9MSb5fCAhXmNNzkeS0fg+zbxeYlztKoDtIRzI/XfL9acvwkrmuReJiWp4uv9zW29j/8nKI
fybhdd/AYrtK4a4ip023ag6Rnd1YsBQ0jLVANFRE9dewvNLfWQDPFj9dEzspvk1FuiC+KR
BVgAtZP1HoB7dns8q40o/1nduMeKusq5IFyCt6J0jJLcd43JhweF1LZeC/ycN8M3aw2iKT
8NgwJewjRAf8EjAAABAQDtKT4LpAn6EC7QAiKAGSVwdqrTiVd5AYcEtKTy4L9Cb1ipZV2d
AmCIjoQjEc9q0i71cIEwsPwNJUI00wCMXRaxcyahQ260hC2N3ZPhZNEws1MPgLBgliTONZ
ZP9VqB4z9DXAz5Y9BFtDyyZQ5S7J8fzHT4gV9vU3pAbXSqircAqQ21UFP0LY/8DZ26x1D
06CduMEJxiQWRsRVA1qzK00XGjs0KxKeMKR0bQ9Ugqyy3yAgvd+XKgmZfGEcF0y32AyzCx
x+DcemKnjELvr/rX+ESiUg/ZZx4Mh4cJLnVEGfg2iAXWc+Abaye135i30dE/BoxsZN51g3
fLho0jDNxagBAAAF2NocmlzYmVyZDIwMDNAZ21haWwuy29tAQID
-----END OPENSSH PRIVATE KEY-----
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/projet1-git (master)
$
```

14) Quatorzième étape : Affiche de la clé publique

```
MINGW32:/c/Users/PC/Desktop/projet1-git
fybhdd/AYrtK4a4ip023ag6Rnd1YsBQ0jLvANfRE9dewwNLfwQDPFj9dEzspvk1Fuic+KR
BVgAtZP1HoB7dns8q40o/1nduMeKusq5IFyCt6J0jJLcd43JhweF1LZeC/ycN8M3aw2ikT
8NgwjewJRAf8EjAAABAQDtKT4LpAn6EC7QAiKAGSVWdqrTiVd5AYcEtkTy4L9Cb1ipZV2d
AmCIjoQjEc9q0i71cIEwsPwNjUI00wCMXRAxcyaHQ260hC2N3ZPhZNEws1MPgLBgl1iTONZ
ZP9VqB4z9DXAz5Y9BFtDyyZQ55R7J8fzHT4gV9vU3pAbXSqircAq021UFPOLY/8DZ26x1D
06CduMEJxiQWRsRVA1qzKXXGjsOKxKeMKrObQ9Ugqyy3yAgvd+XKgmZfGecF0y32AyzCx
x+DcemKnjELvr/rX+ESiUg/ZZx4Mh4cJLnVEGfg2iAXWc+Abaye135i30dE/BoxsZN51g3
fLho0jDNxagBAAAF2NocmlzYmVy dIwMDNAZ21haWwY29tAQID
-----END OPENSSH PRIVATE KEY-----

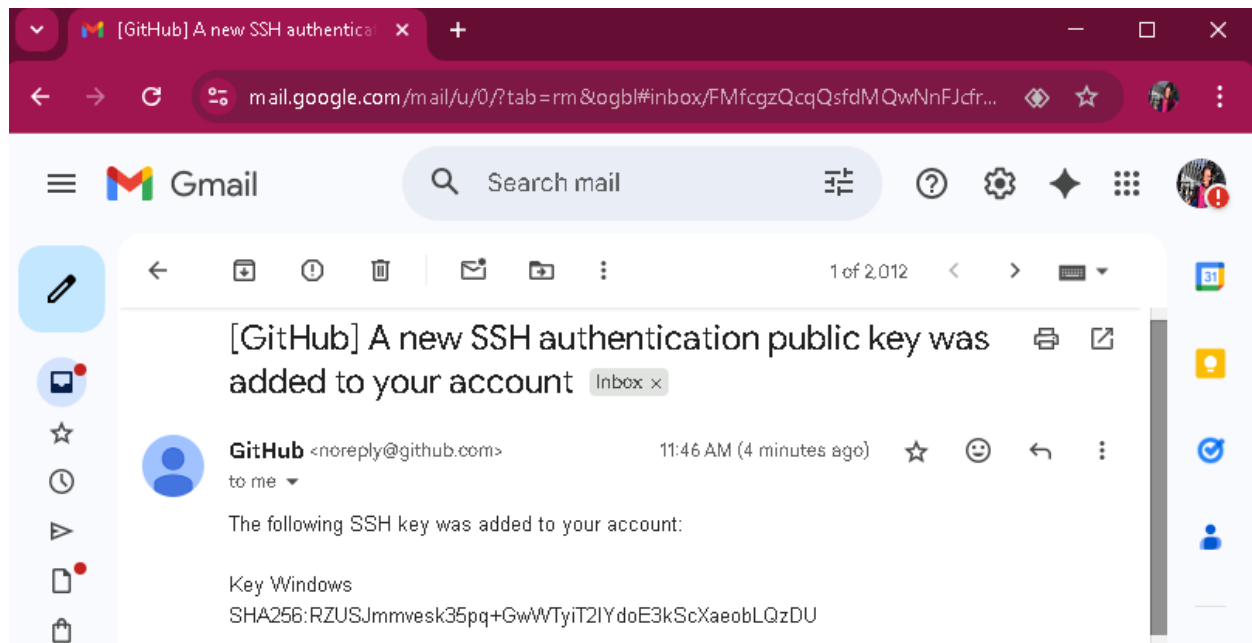
PC@DESKTOP-U5501MB MINGW32 ~/Desktop/projet1-git (master)
$ AC

PC@DESKTOP-U5501MB MINGW32 ~/Desktop/projet1-git (master)
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDM8Hy1cwa1C13/ZYzZik/s3Z1d9F1prnzNp3vUYrEL
RwhVNCza0TB9fCViBY4MWhlyRWLFUg1NzDi2hywJOMZwohscJTunMdv282iGMOaXHj1fEViy8WQWJctB
ifIDEfsEZsnX3mrkaNOM/jOVpNiNSQs783s1j4Z9yB3CZiJmN7wRtXWCZhI3gYN9kkWuDv/owurTtJEz
bMvDb2YagnehE/4RKBEZfz005jKf/PfHGIwg7GGsGIaViorVH2gfi7q3RatUzbh1fIStaZCzj4Heve1

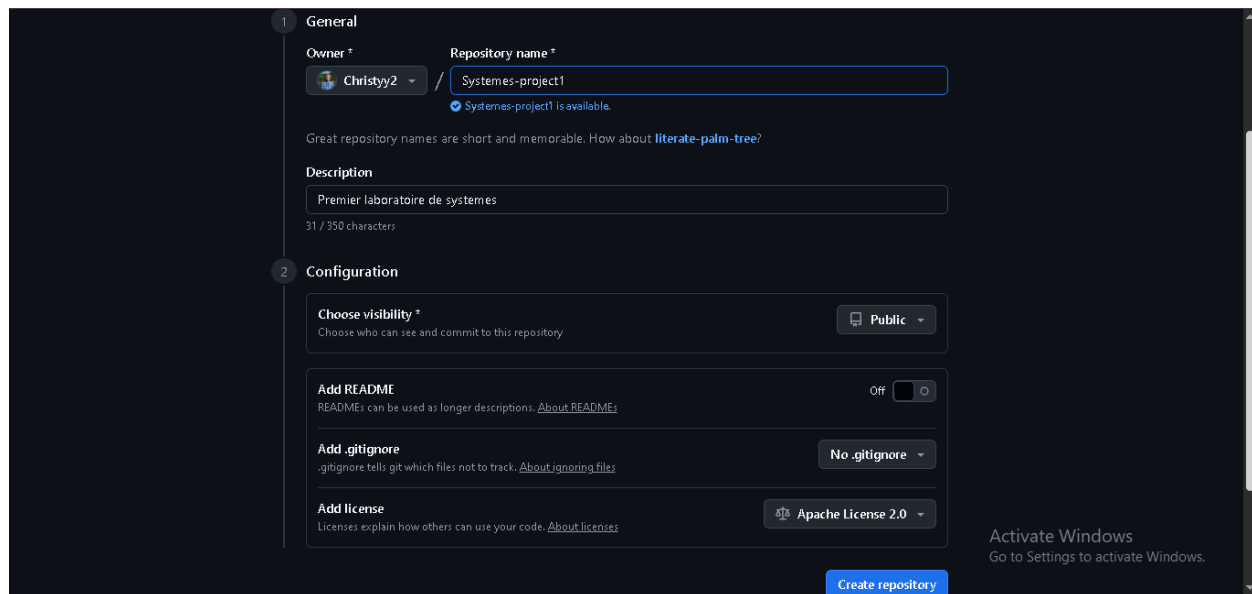
PC@DESKTOP-U5501MB MINGW32 ~/Desktop/projet1-git (master)
$
```

15) Quinzième étape : Ajout réussie de la clé sur mon compte GitHub

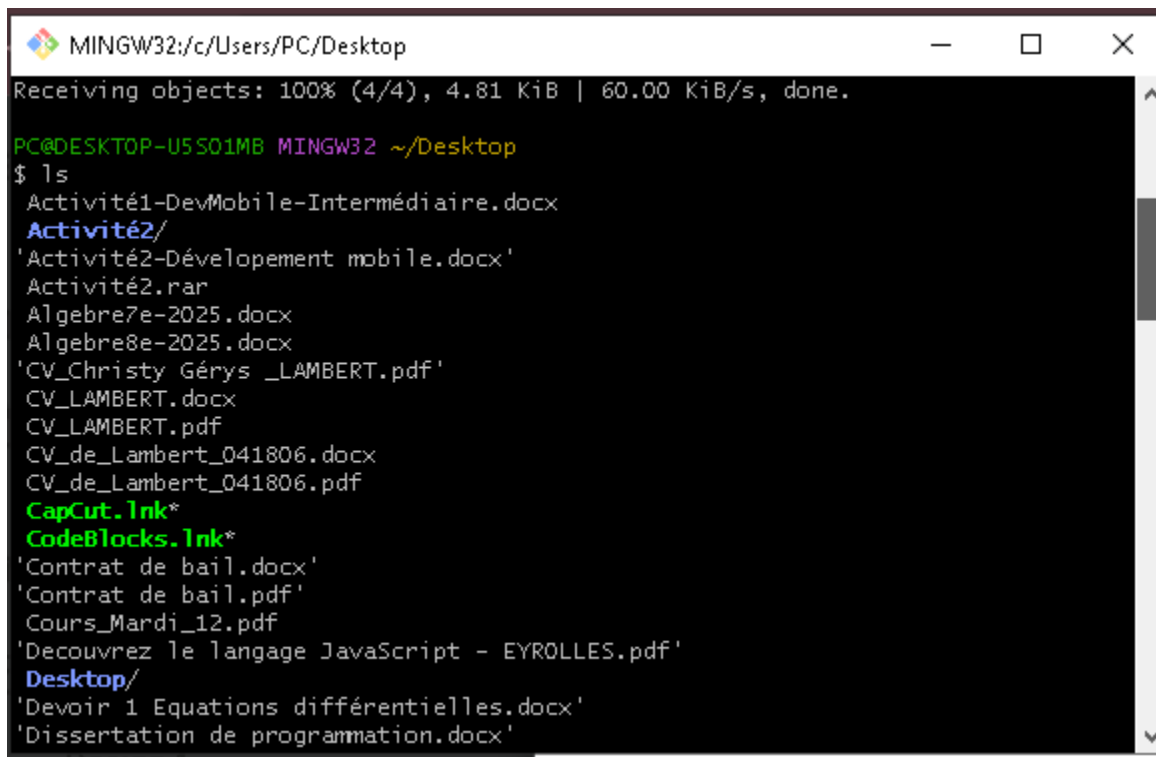
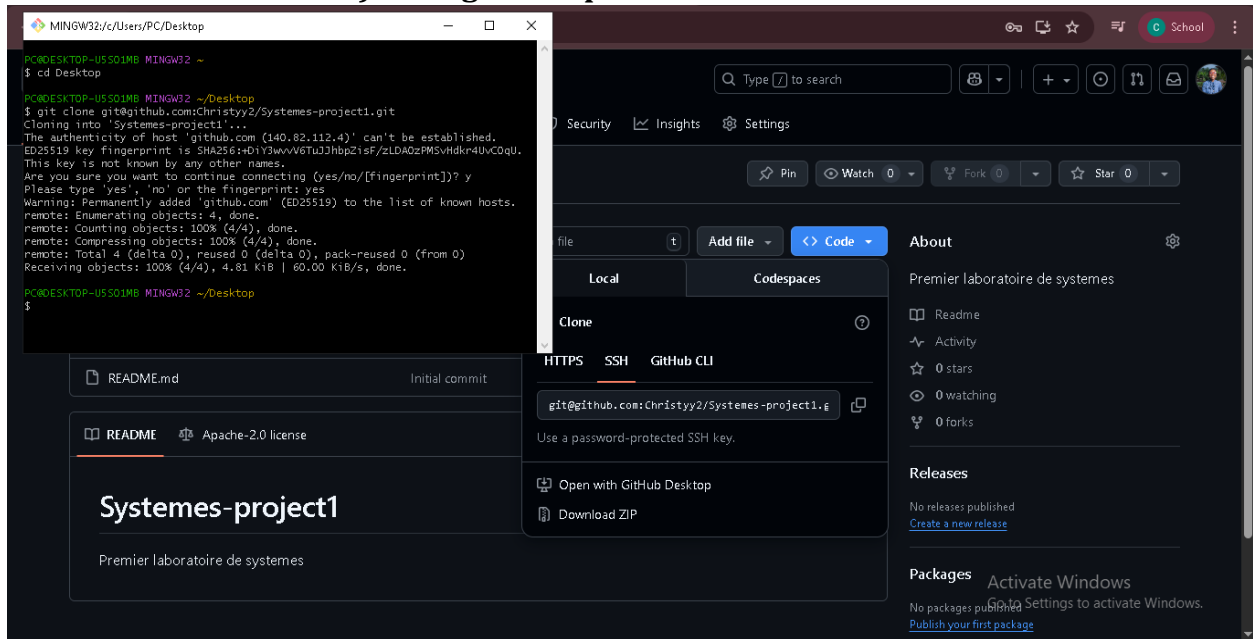
The screenshot shows the GitHub 'Settings' page for the user 'Christy Gérys LAMBERT (Christyy2)'. The 'SSH and GPG keys' tab is selected in the left sidebar. A notification at the top states: 'You have successfully added the key 'Key Windows''. The 'SSH keys' section displays a list of keys. One key is listed: 'Key Windows' with a SHA256 hash 'SHA256:RZUSJmmvesk35pq+GwITy1T21YdoE3kScXaeobLQzDU', added on 'Oct 17, 2025', and marked as 'Never used'. A 'Delete' button is next to it. Below the list, there is a link to a guide: 'Check out our guide to [connecting to GitHub using SSH keys](#) or [troubleshoot common SSH problems](#).' The 'GPG keys' section below it states: 'There are no GPG keys associated with your account.' At the bottom right, there is a 'New GPG key' button and a 'Activate Windows' watermark.



16) Seizième étape : Création du dépôt GitHub



17) Clonage du dépôt GitHub via la clé SSH



MINGW32:/c/Users/PC/Desktop/Systemes-project1

'~\WRL2995.tmp'

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop

\$ cd Systemes-project1

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/Systemes-project1 (main)

\$ echo "Systeme d'exploitation!" > Module.txt

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/Systemes-project1 (main)

\$ git add

Nothing specified, nothing added.

hint: Maybe you wanted to say 'git add .?'

hint: Disable this message with "git config advice.addEmptyPaths false"

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/Systemes-project1 (main)

\$ git add .

warning: in the working copy of 'Module.txt', LF will be replaced by CRLF the next time Git touches it

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/Systemes-project1 (main)

\$ git commit -m "Ajout Module"

[main d644723] Ajout Module

1 file changed, 1 insertion(+)

create mode 100644 Module.txt

MINGW32:/c/Users/PC/Desktop/Systemes-project1

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/Systemes-project1 (main)

\$ git commit -m "Ajout Module"

[main d644723] Ajout Module

1 file changed, 1 insertion(+)

create mode 100644 Module.txt

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/Systemes-project1 (main)

\$ git branch -M main

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/Systemes-project1 (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), 338 bytes | 338.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To github.com:Christyy2/Systemes-project1.git

1654b9e..d644723 main -> main

branch 'main' set up to track 'origin/main'.

PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/Systemes-project1 (main)

\$ |

SECOND EXERCICE

- ✓ Création d'un dépôt GitHub nommé TD

The screenshot shows the GitHub repository creation interface. The 'General' tab is active, showing the repository name 'TD' and the owner 'Christy2'. The description is 'Premier Laboratoire du cours Systemes'. The 'Configuration' tab is also visible, showing options for visibility (Public), adding a README, .gitignore, and a license (Apache License 2.0).

1 General

Owner * Repository name *

Christy2 / TD

TD is available.

Great repository names are short and memorable. How about [supreme-octo-goggles?](#)

Description

Premier Laboratoire du cours Systemes

37 / 350 characters

2 Configuration

Choose visibility * Choose who can see and commit to this repository

Public

Add README READMEs can be used as longer descriptions. [About READMEs](#)

On

Add .gitignore .gitignore tells git which files not to track. [About ignoring files](#)

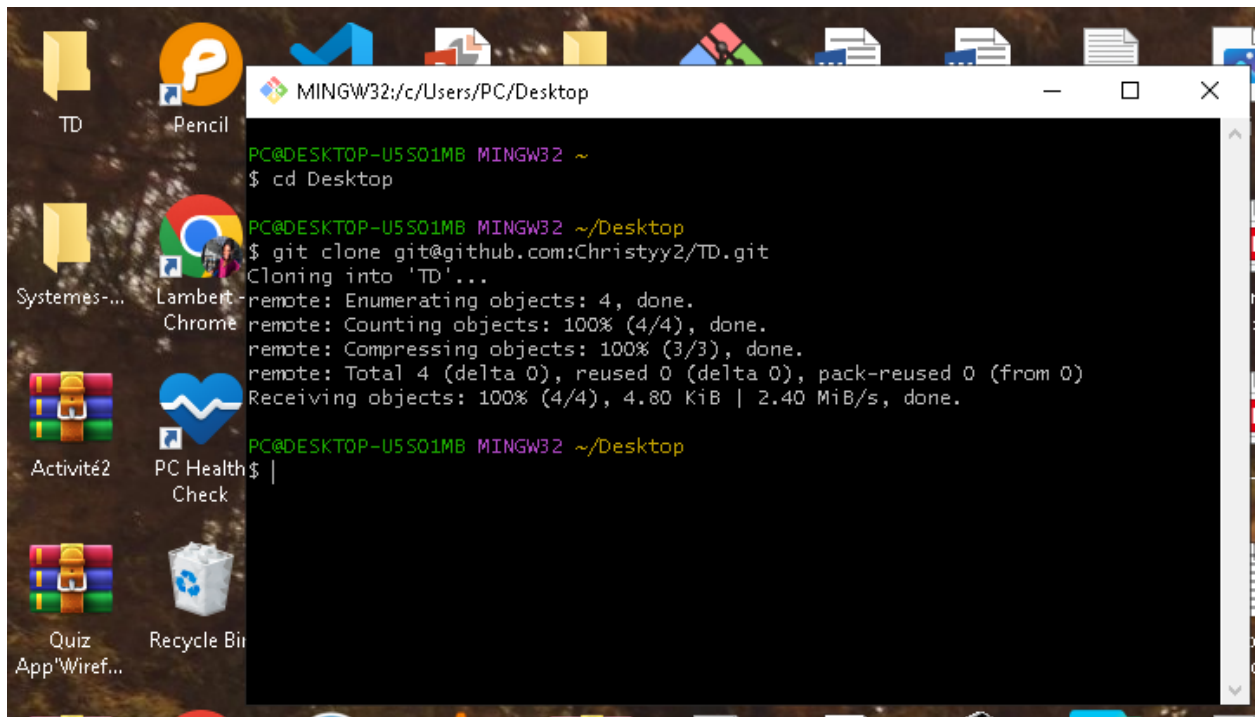
No .gitignore

Add license Licenses explain how others can use your code. [About licenses](#)

Apache License 2.0

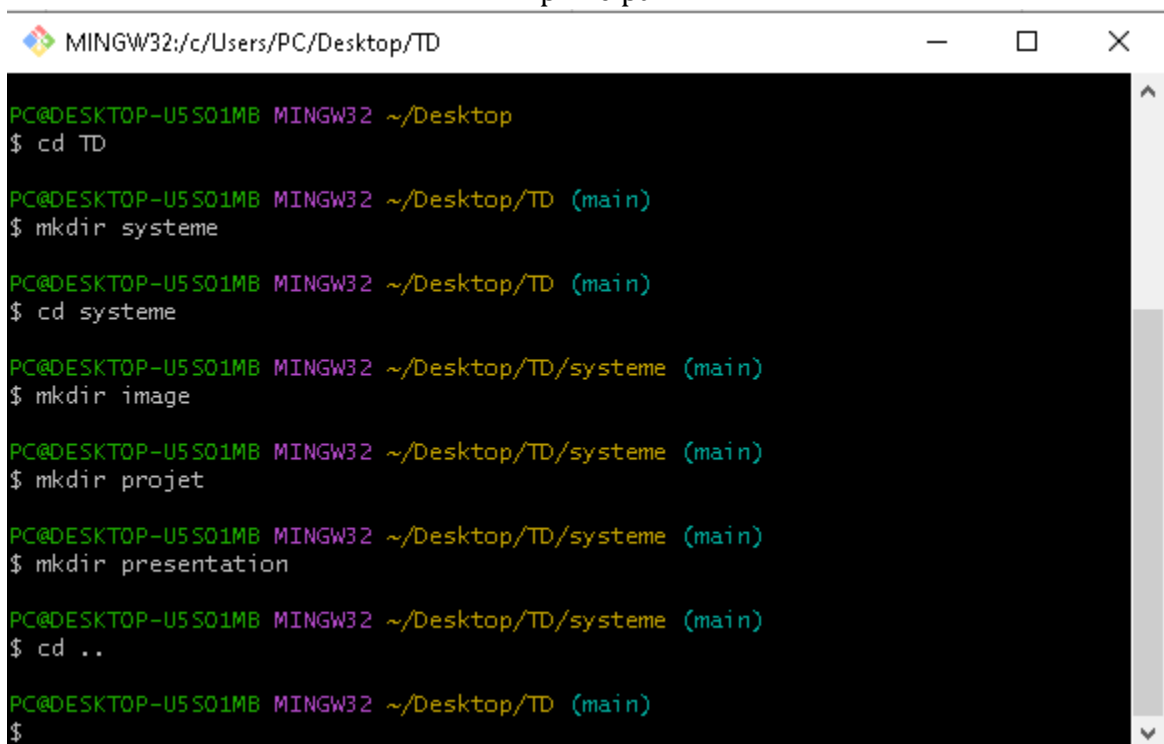
Activ
Go to S

- ✓ Clonage du dossier TD sur mon bureau



```
MINGW32:/c/Users/PC/Desktop
PC@DESKTOP-U5S01MB MINGW32 ~
$ cd Desktop
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ git clone git@github.com:Christyy2/TD.git
Cloning into 'TD'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), 4.80 KiB | 2.40 MiB/s, done.
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ |
```

- ✓ Création des dossiers systeme, image, projet, presentation dans le dossier principal TD



```
MINGW32:/c/Users/PC/Desktop/TD
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop
$ cd TD
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/TD (main)
$ mkdir systeme
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/TD (main)
$ cd systeme
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/TD/systeme (main)
$ mkdir image
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/TD/systeme (main)
$ mkdir projet
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/TD/systeme (main)
$ mkdir presentation
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/TD/systeme (main)
$ cd ..
PC@DESKTOP-U5S01MB MINGW32 ~/Desktop/TD (main)
$
```

Conclusion

Notions acquises durant ce laboratoire :

- ✓ Familiarisation avec les termes Git (dépôt local) et GitHub (dépôt distant)
- ✓ Capacité à héberger un projet sur GitHub et à cloner un projet sur notre ordinateur avec git
- ✓ Utilisation de certaines commandes git et leurs utilisations (et d'autres termes techniques) :
 1. git --version : Permet de vérifier la version git installée sur son ordinateur.
 2. git config --global user.name et git config --global user.email : Permet de configurer le compte GitHub avec des données utilisateurs.
 3. git config --list : Permet d'afficher la liste des configurations de Git.
 4. ls : permet l'affichage de tous les répertoires.
 5. cd : Permet l'ouverture d'un dossier.
 6. mkdir : Permet de créer un dossier.
 7. git init : Permet l'initialisation d'un dépôt.
 8. README.me : Fichier texte écrit en Markdown.
 9. git log : Permet d'afficher l'historique des **commits** dans un dépôt git.
 10. git status : Sert à afficher l'état actuel du dépôt Git.
 11. git add . : Sert à ajouter des fichiers
 12. git commit -m : Valide les changements ajoutés.
 13. git branch -M main : Permet de renommer la branche actuelle en main, même si une branche main existe déjà.
 14. git push -u origin main : Permet d'envoyer la branche main vers GitHub.

