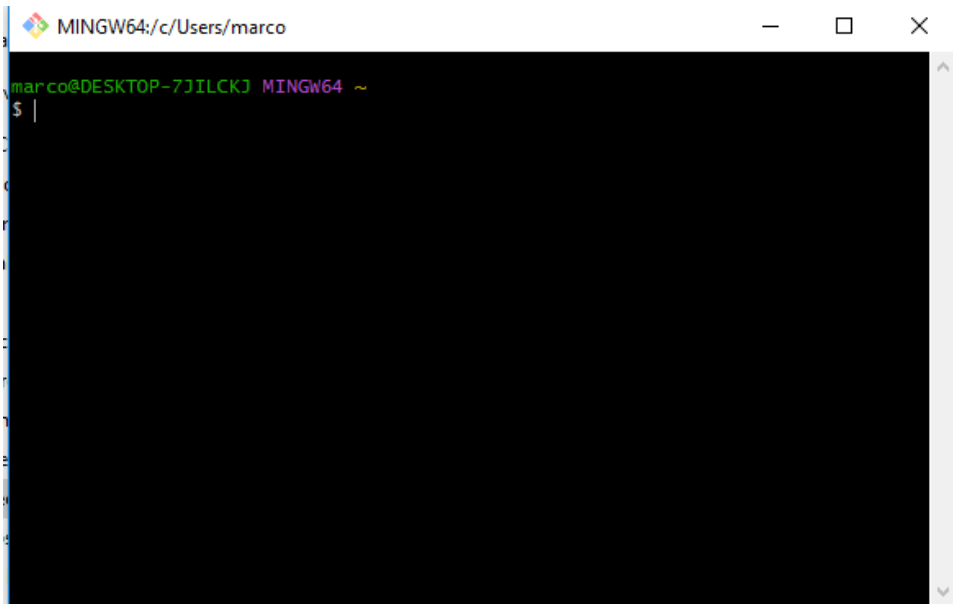


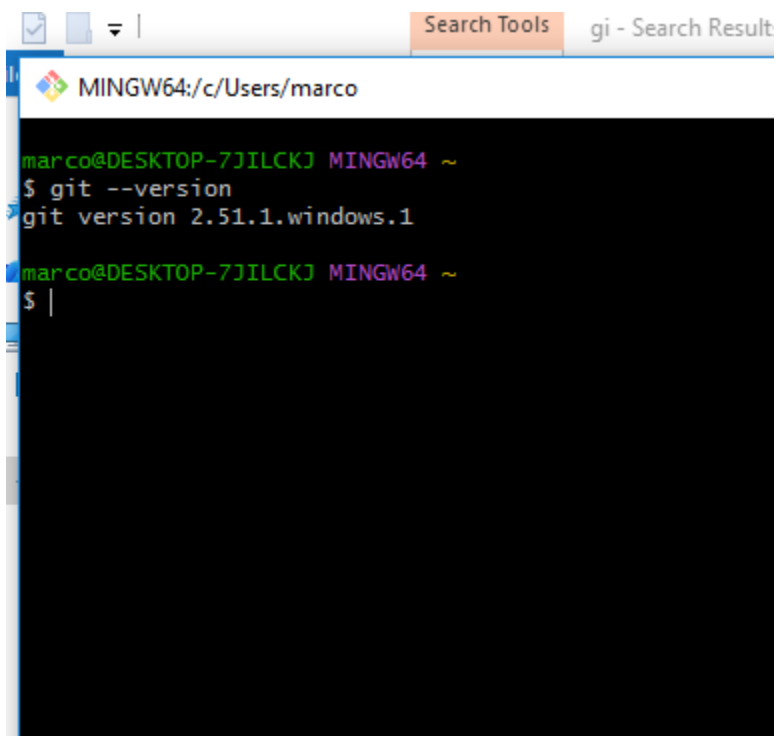
1) Installation de Git de Git sur Windows



```
MINGW64: c:/Users/marco  
marco@DESKTOP-7JILCKJ MINGW64 ~  
$ |
```

2) Verification de l'installation

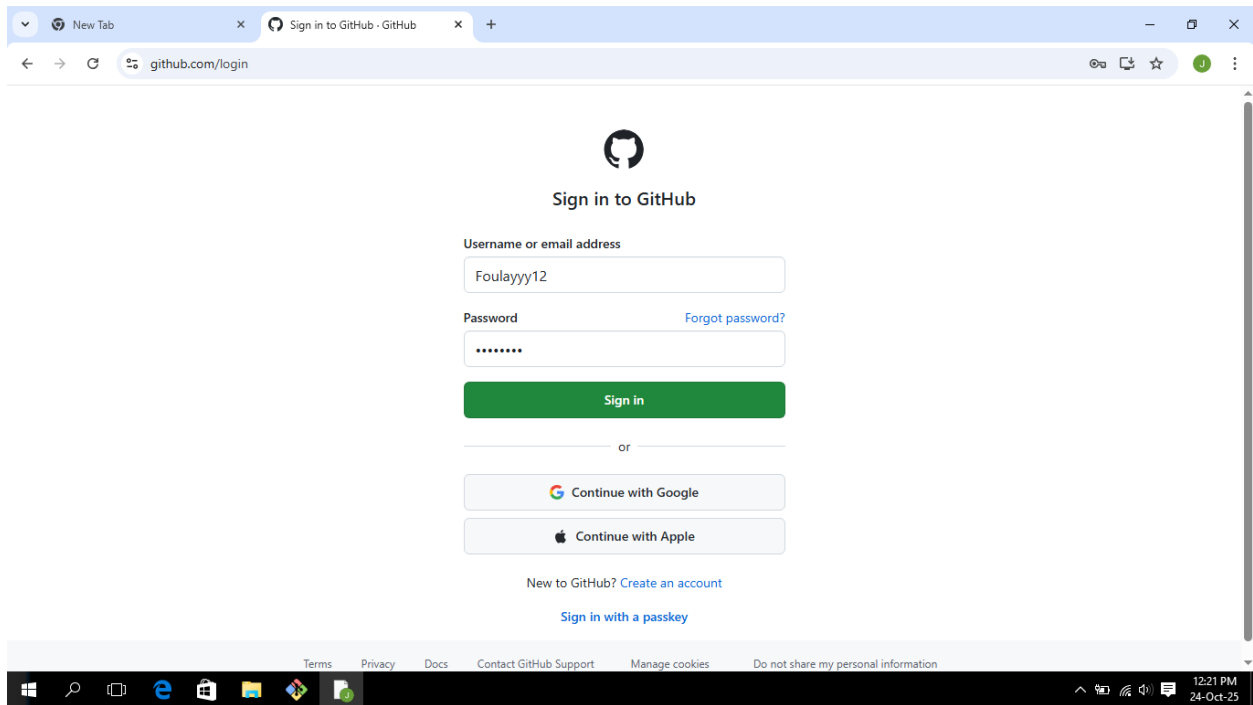
on verifie si Git fonctionne en ajoutant git --version dans Gith puis on fait Enter



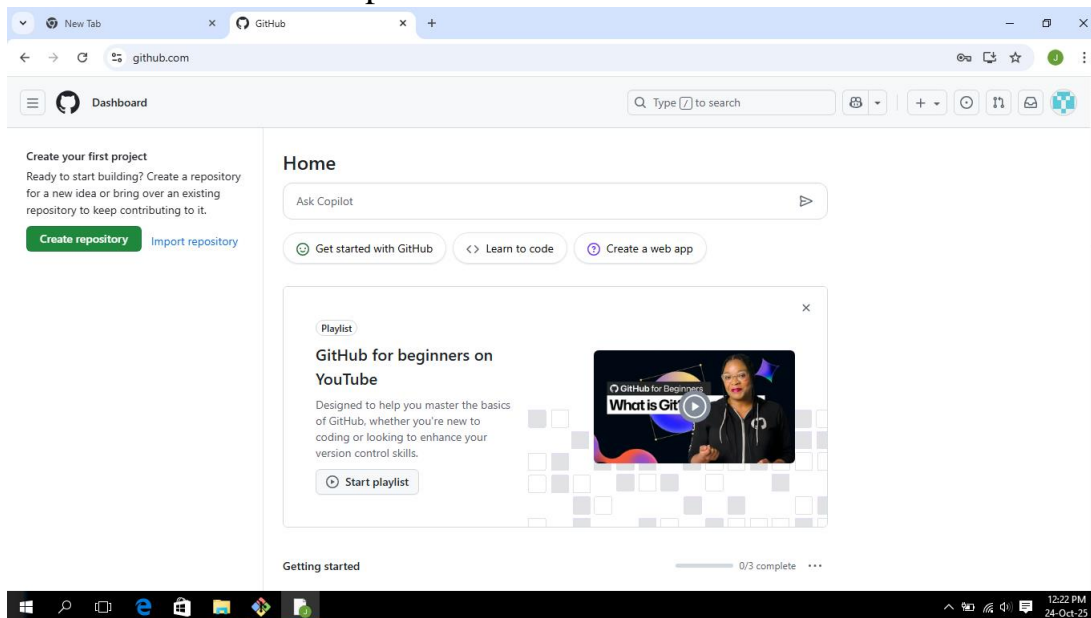
```
MINGW64: c:/Users/marco  
marco@DESKTOP-7JILCKJ MINGW64 ~  
$ git --version  
git version 2.51.1.windows.1  
marco@DESKTOP-7JILCKJ MINGW64 ~  
$ |
```

3) Creation du compte Github

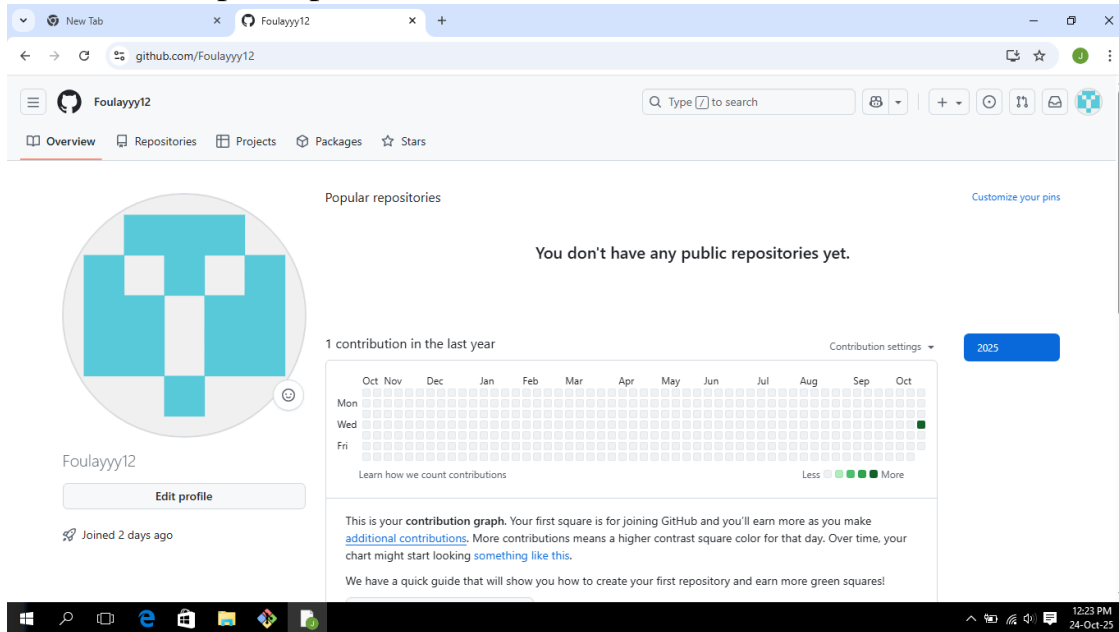
Pour créer le compte on monte sur chrome et écrit Github puis on procede à l'inscription en ajoutant votre email, mot de passe et un nom d'utilisateur pour le compte Github après on choisit le pays (Haiti), apres on fait créer une compte



On ouvre Github pour arriver sur le Dashboard



On ouvre son profil pour verifier



4) Configuration de Git

on commence par mettre git config --global user.name "Foulayyy12"

ou git config --global user.email francoisjuliencw@gmail.com

Ensuite pour verifier on fait git config --list

```
MINGW64:/c/Users/marco
git version 2.51.1.windows.1
marco@DESKTOP-7JILCKJ MINGW64 ~
$ git config --global user.name "Foulayyy12"
marco@DESKTOP-7JILCKJ MINGW64 ~
$ git config --global user.email "francoisjuliencw@gmail.com"
marco@DESKTOP-7JILCKJ 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=Foulayyy12
user.email=francoisjuliencw@gmail.com
```

5) Teste de Git avec Powershell

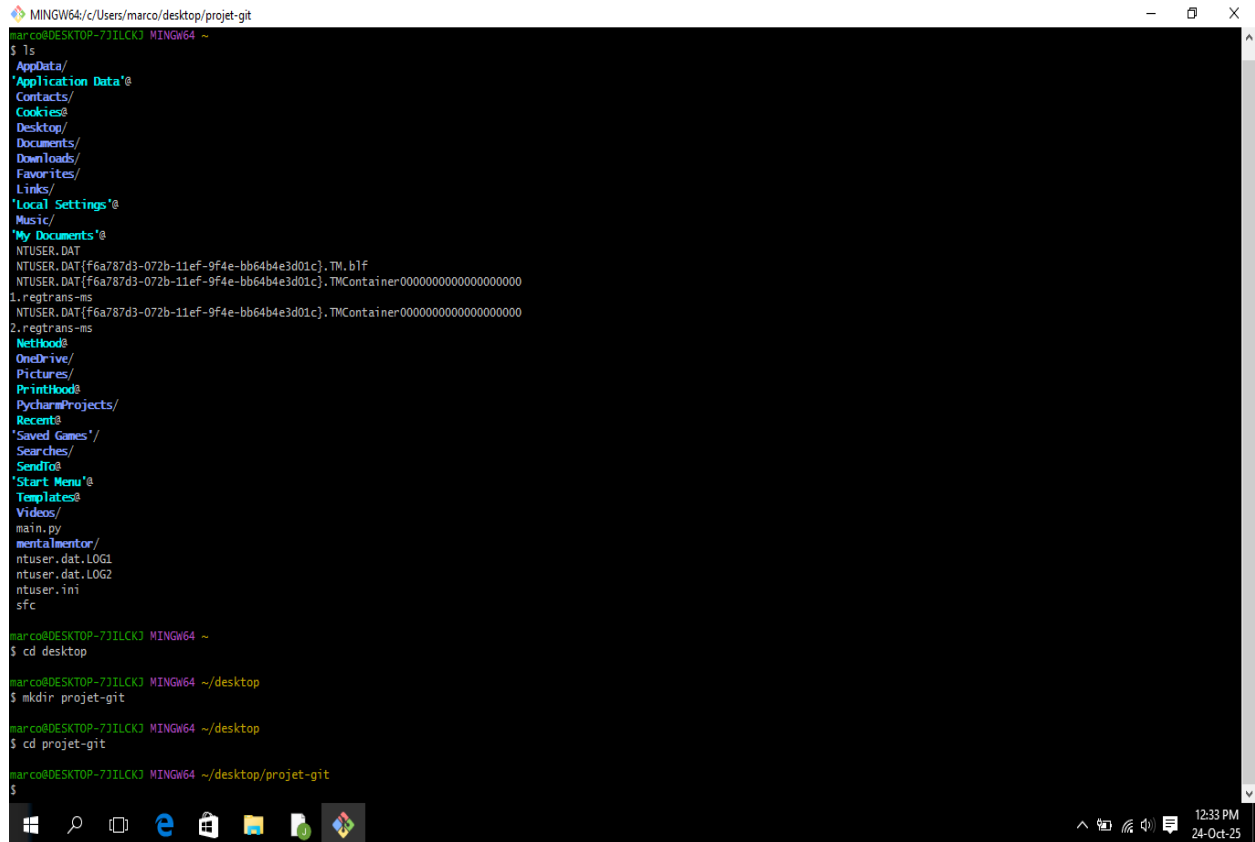
Pour créer un dossier on fait

ls

cd Desktop

mkdir projet-git

cd projet-git



```
mingw64/c/Users/marco/desktop/projet-git
marco@DESKTOP-7JILCKJ MINGW64 ~
$ ls
AppData/
'Application Data'@
Contacts/
Cookies@
Desktop/
Documents/
Downloads/
Favorites/
Links/
'Local Settings'@
Music/
'My Documents'@
NTUSER.DAT
NTUSER.DAT{f6a787d3-072b-11ef-9f4e-bb64b4e3d01c}.TM.b1f
NTUSER.DAT{f6a787d3-072b-11ef-9f4e-bb64b4e3d01c}.TMContainer000000000000000000
1.regtrans-ms
NTUSER.DAT{f6a787d3-072b-11ef-9f4e-bb64b4e3d01c}.TMContainer000000000000000000
2.regtrans-ms
NetHood@
OneDrive/
Pictures/
PrintHood@
PycharmProjects/
Recent@
'Saved Games'/
Searches/
SendTo@
'Start Menu'@
Templates@
Videos/
main.py
mentalmirror/
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
sfc
marco@DESKTOP-7JILCKJ MINGW64 ~
$ cd desktop

marco@DESKTOP-7JILCKJ MINGW64 ~/desktop
$ mkdir projet-git

marco@DESKTOP-7JILCKJ MINGW64 ~/desktop
$ cd projet-git

marco@DESKTOP-7JILCKJ MINGW64 ~/desktop/projet-git
$
```

Pour initialiser le depot on fait git init

Pour créer un fichier on fait

echo "Hello git" >README . md

git add README . md

git commit -m " Premier commit avec Powershell"

git status

```
MINGW64/c/Users/marco/desktop/projet-git
sfc

marco@DESKTOP-73ILCKJ MINGW64 ~
$ cd desktop

marco@DESKTOP-73ILCKJ MINGW64 ~/desktop
$ mkdir projet-git

marco@DESKTOP-73ILCKJ MINGW64 ~/desktop
$ cd projet-git

marco@DESKTOP-73ILCKJ MINGW64 ~/desktop/projet-git
$ git int
git: 'int' is not a git command. See 'git --help'.

The most similar command is
  init

marco@DESKTOP-73ILCKJ MINGW64 ~/desktop/projet-git
$ git init
Initialized empty Git repository in C:/Users/marco/Desktop/projet-git/.git/

marco@DESKTOP-73ILCKJ MINGW64 ~/desktop/projet-git (master)
$ echo "Hello Git" > README.md

marco@DESKTOP-73ILCKJ MINGW64 ~/desktop/projet-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

marco@DESKTOP-73ILCKJ MINGW64 ~/desktop/projet-git (master)
$ git commit -m "Premier commit avec PowerShell"
[master (root-commit) 5722d93] Premier commit avec PowerShell
1 file changed, 1 insertion(+)
create mode 100644 README.md

marco@DESKTOP-73ILCKJ MINGW64 ~/desktop/projet-git (master)
$ git status
On branch master
nothing to commit, working tree clean

marco@DESKTOP-73ILCKJ MINGW64 ~/desktop/projet-git (master)
$ git log
commit 5722d935917effbe4b4c905dce479a24fcb9ba35 (HEAD -> master)
Author: Foulayy12 <francoisjuliencw@gmail.com>
Date: Fri Oct 24 12:39:44 2025 -0400

    Premier commit avec PowerShell

marco@DESKTOP-73ILCKJ MINGW64 ~/desktop/projet-git (master)
$
```

6) Gestion d'une clé SSH

on fait la commande `ssh-keygen -t rsa -b 4096 -C "votre.email.com"`

```
MINGW64/c/Users/marco/Desktop

marco@DESKTOP-73ILCKJ MINGW64 ~
$ cd Desktop

marco@DESKTOP-73ILCKJ MINGW64 ~/Desktop
$ ssh-keygen -t rsa -b 4096 -C "francoisjuliencw@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c:/Users/marco/.ssh/id_rsa):
/c:/Users/marco/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase for "/c:/Users/marco/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c:/Users/marco/.ssh/id_rsa
Your public key has been saved in /c:/Users/marco/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:11NC1NC30K3G4s4MLvF1Xs1p7mnet1geTysF2D8 francoisjuliencw@gmail.com
The key's randomart image is:
+--[RSA 4096]-----
|
|.o+
|..+..
|o=.o
|..ooo..So
|..+..o
|o..oo+ o.o.o|
|+o..oo+ow.E|
|..+..o..+..o|
+-----[SHA256]-----

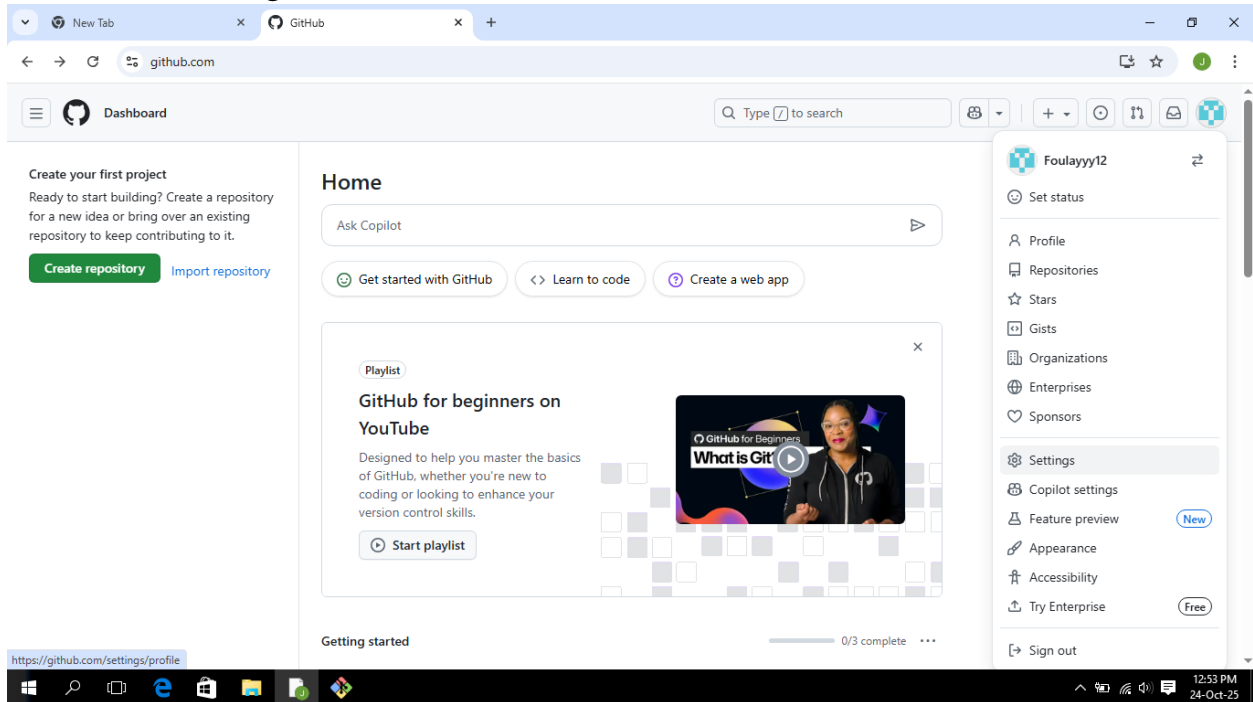
marco@DESKTOP-73ILCKJ MINGW64 ~/Desktop
$ cat ~/.ssh/id_rsa
cat: /c:/Users/marco/: Is a directory
cat: .ssh/id_rsa: No such file or directory

marco@DESKTOP-73ILCKJ MINGW64 ~/Desktop
$ cat ~/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzC1ZKtdjEAAAABG5vbmUAAAABm9uZQAAAAAAAAAAAAAAACFwAAAAZcc2gtcn
NHAIAAAAEAAQAAAGAEAlPhnCmqAE1Dna3enCXAcCOENSRQVPBORTNYTOPgm0cnKbs+NieZ
B4RCoQZARWkYtOf08/gnc3TCBj5JXZE+gVK5FOWjxZp7nqoi1va3urPpP385tun80t/Jd
Sc7469evY/7Dmwa7Tdx8721d36wP/byz3DZC15odBwTTMEqdc6ShH42o1cwwOP
Kk3VKsKkZwTw65OPQ0xvSw/KF1Y57H1np9K5w1I9Z1SGCotw8gQvZFT9UFDX6t6xhXEA
1X4E+V6DKjE1328pG8+7VFj0Tufkeez21phRtQMSKRMuxnMYH87au4ys098+9hgXa3wp
hceInBPJR8AmH3czDPfne1CQ0Z4E3PDXK8neBkGmstWvKXP65dv3L5HE264hds
S8FveVbeVAtgwPM+q140C80tLrRTub0axzj9o1xyYtbfF542Yw8/K1NBFRUF52hJKL
D9FF1H3j5IPjpm0d0F64752o2zCMCK563H8VtqD2QbuzrTTUxyxoGwRBRiWpuTr0uH2J
GyXPTfBVp7ppjctFvNacbvyskctZYv8F530TE43Dr8Q0b6bchp/oi+xx88R/h/WGsqm1rc
DJrV9AQZqL/Z0H13w15q38pK5BmXKcFHB53N6/G3ceXQ0c9490w491Frdq15L0
EAAADQ94ekTvehpEAAAHC3NoLXJzYQAAAGAEAlPhnCmqAE1Dna3enCXAcCOENSRQVPB0
RtNYTOPgm0cnKbs+NieZB4RCoQZARWkYtOf08/gnc3TCBj5JXZE+gVK5FOWjxZp7nqoi1v
```

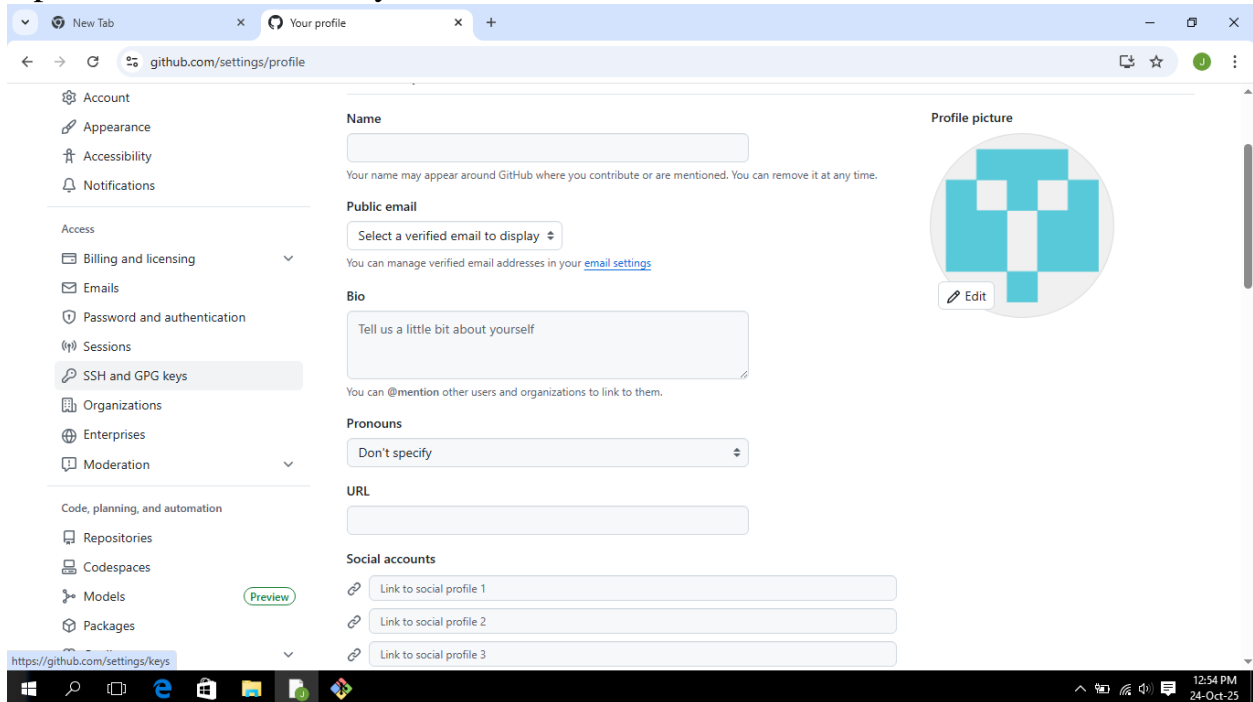
Pour rendre publique la cle on fait
`cat ~/.ssh/id_rsa`

Puis on copie la clé afficher dans la terminale, puis on suit ce processus en montant sur votre compte Github

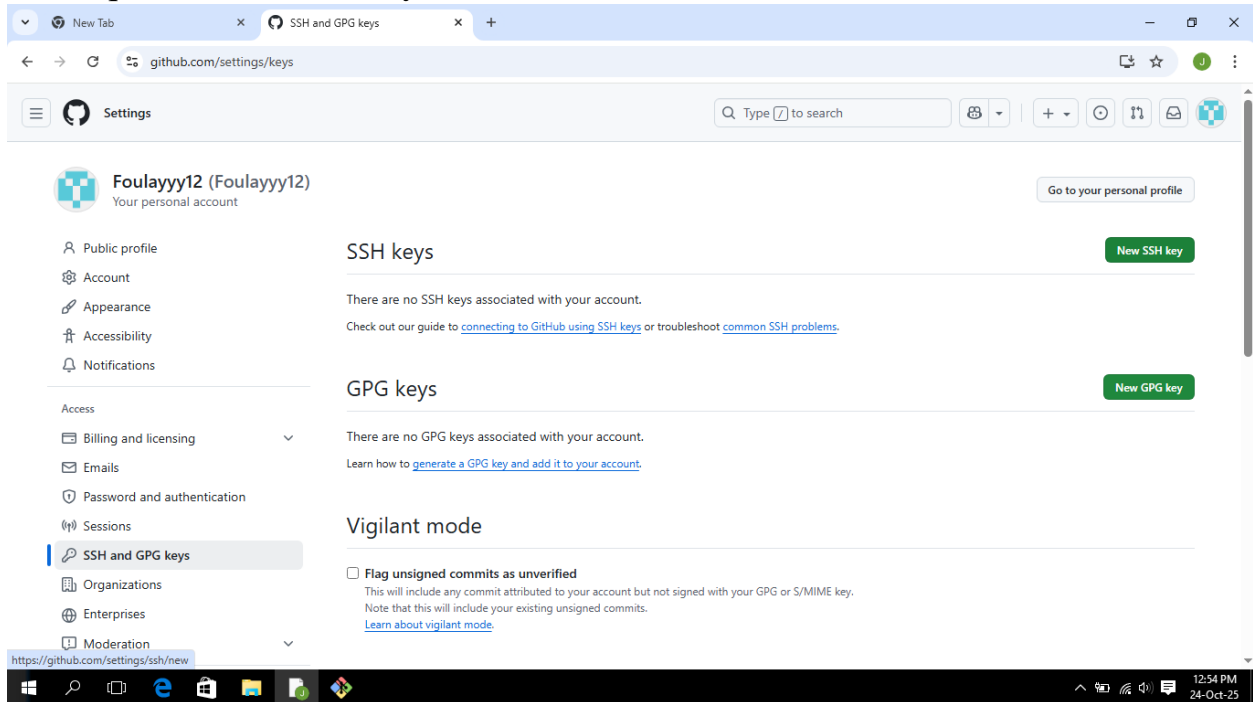
On va dans settings



Après SSH and GPG Key

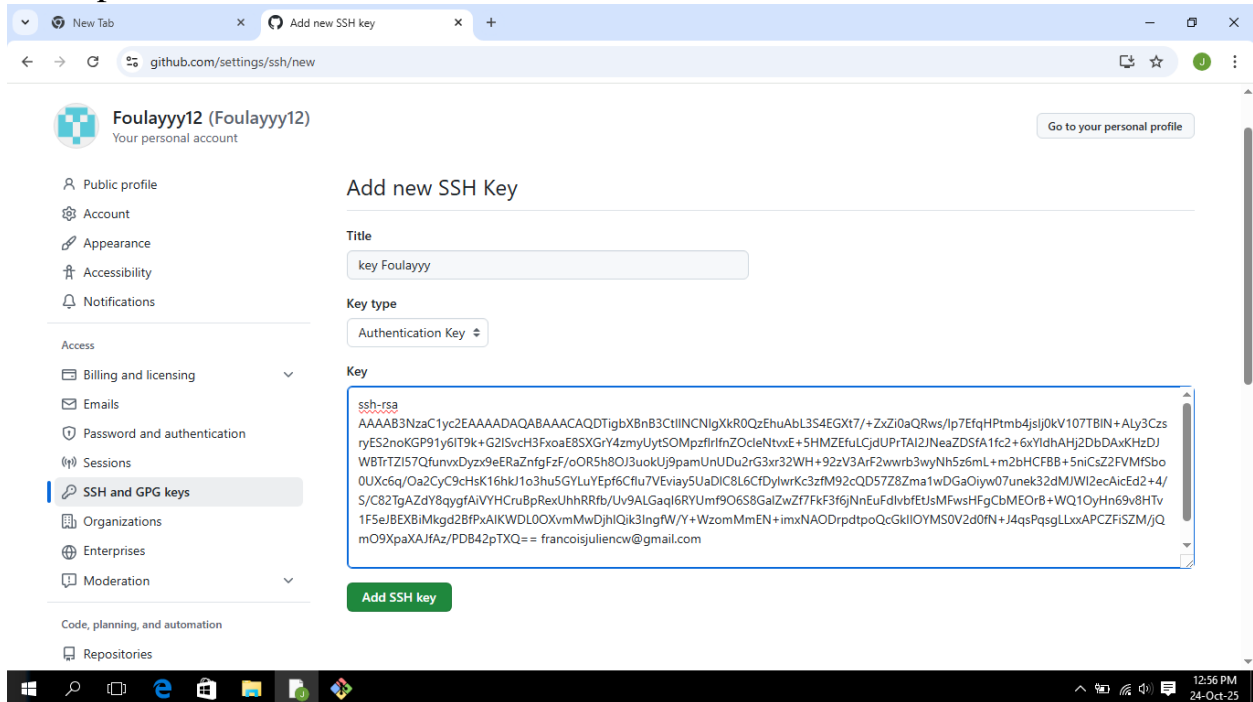


On clique sur New SSh key

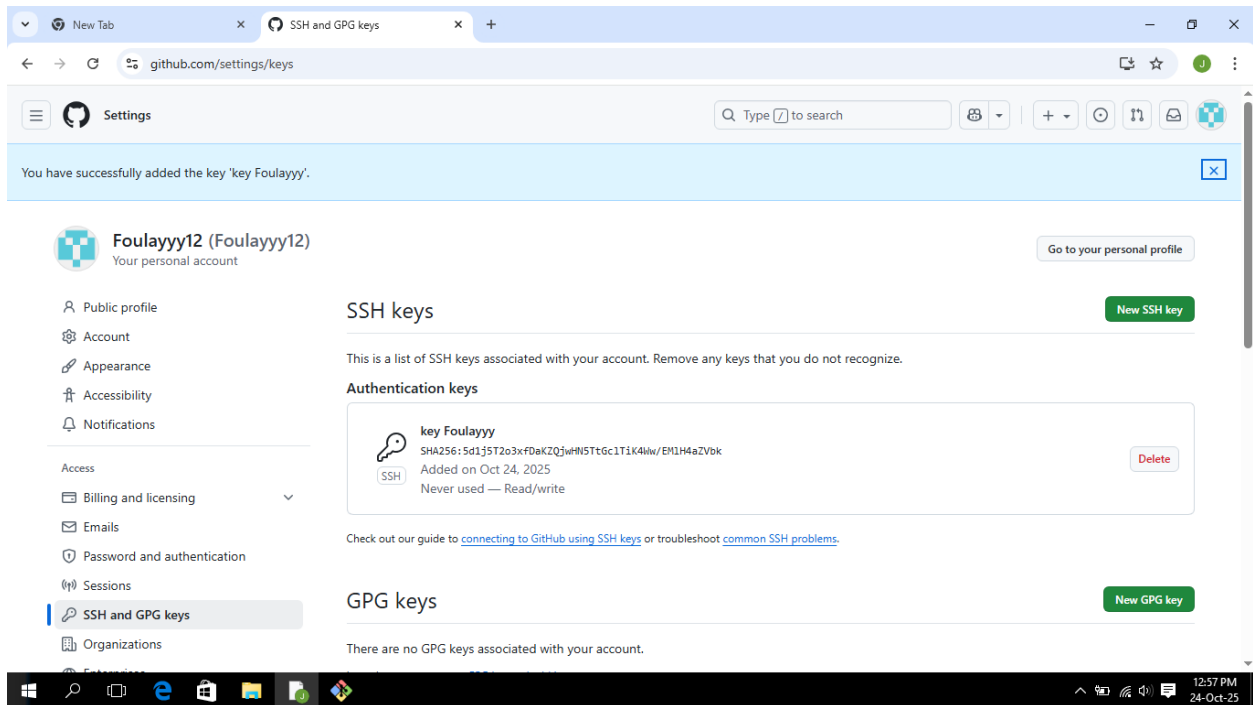


The screenshot shows the GitHub Settings page for the user 'Foulayyy12'. The left sidebar contains navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and licensing, Emails, Password and authentication, Sessions, SSH and GPG keys (selected), Organizations, Enterprises, and Moderation. The main content area is titled 'SSH keys' and 'GPG keys'. The 'SSH keys' section has a 'New SSH key' button and a message: 'There are no SSH keys associated with your account. Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).' The 'GPG keys' section has a 'New GPG key' button and a message: 'There are no GPG keys associated with your account. Learn how to [generate a GPG key and add it to your account](#).' The 'Vigilant mode' section has a checkbox for 'Flag unsigned commits as unverified' and a message: 'This will include any commit attributed to your account but not signed with your GPG or S/MIME key. Note that this will include your existing unsigned commits. [Learn about vigilant mode](#).'

On copie la clé

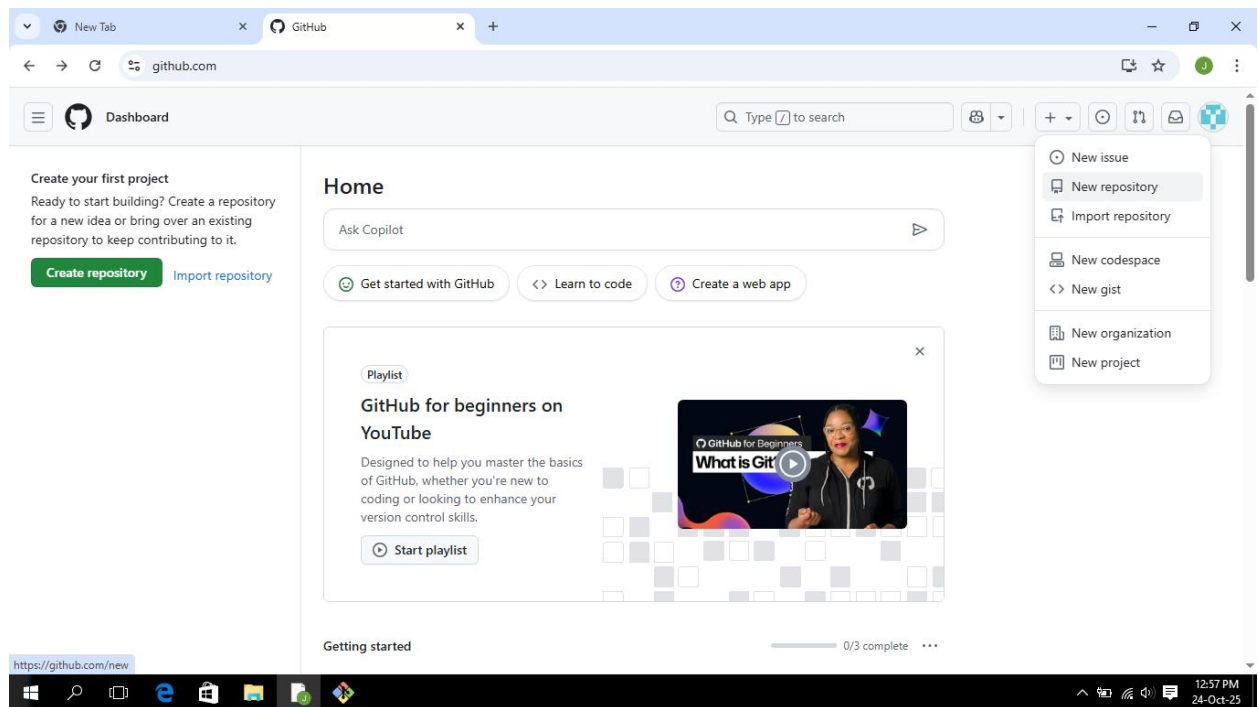


The screenshot shows the GitHub Settings page for the user 'Foulayyy12' with the 'Add new SSH Key' form open. The 'Title' field is 'key Foulayyy'. The 'Key type' is 'Authentication Key'. The 'Key' field contains a long SSH key string: `ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCTIgbX8nB3CtIIINCNIgXkR0QzEhuAbL3S4EGXt7/+ZxZi0aQRws/lp7EfqHPtmb4jsj0kV107TBIN+AlY3CzsyES2nokGP91y6IT9k+G2ISvcH3FxoE8SXGrY4zmyUytSOMpzflrfnZOcleNtvxE+5HMEZfuLCjdUPrTAI2JNeaZDSfA1fc2+6xYldhAHj2DbDaxHzDJWBTrTZIS7QfunvxDyzx9eERaZnfgFzF/oORS5h8OJ3uokUj9pamUnUDu2rG3xr32WH+92zV3ArF2wwrb3wyNh5z6mL+m2bHCFBB+5niCsZ2FVMf5bo0UXc6q/Oa2Cy9cHsk16hkl1o3hu5GYLuYEp6Cflu7VEviay5UaDIC8L6CfDylwrKc3zfM92cQD57Z8Zma1wDGaOiyw07unek32dMJWL2ecAicEd2+4/S/C82TgAZdY8yqgYfAVYHCru8pRexUhhRRRfb/Uv9ALGaql6RYUm9O6S8GalZwZ77FkF3f6jNnEuFdlvbfEtUsMFwSHFgCbMEOrB+WQ1OyHn69v8HTv1F5eJBEXBilMkgd2BfPxAIKWDL0OXvmMwDjhIqik3IngfW/Y+WzomMmEN+imxNAODrpdtpoQcGkII0YMS0V2d0fN+J4qsPqsgLLxxAPCFISZM/jQmO9XpaXAfaz/PDB42pTXQ== francoisjulienwc@gmail.com`. The 'Add SSH key' button is visible at the bottom.

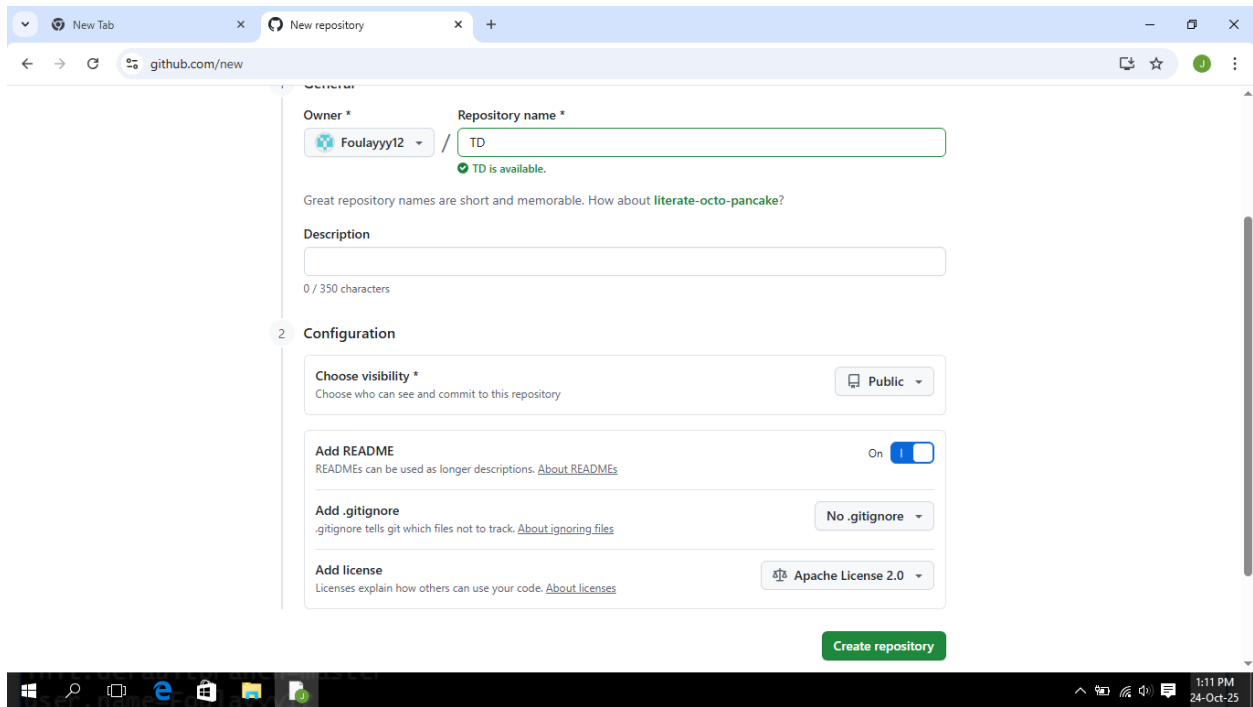


7) Creation d'un depot GitHub

On va dans Github on clique sur Sur New repository et on rempli les informations

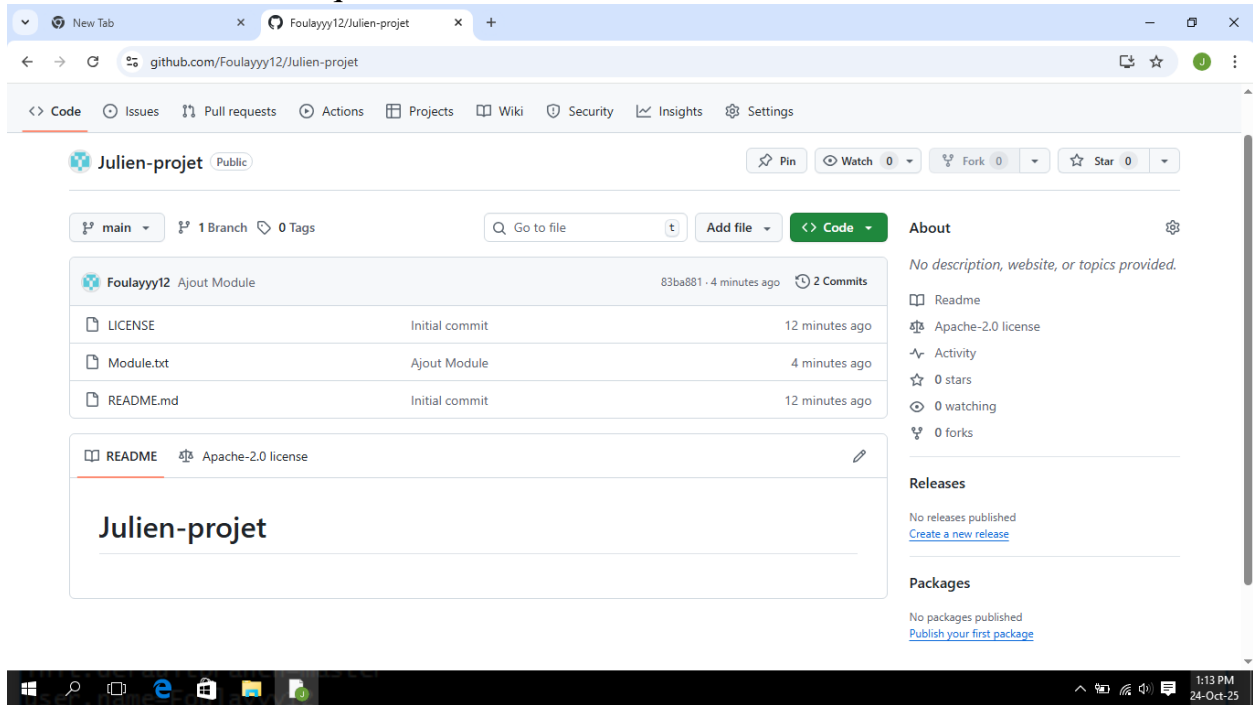


On rend le compte publique, ajoute une license puis on crée le repository

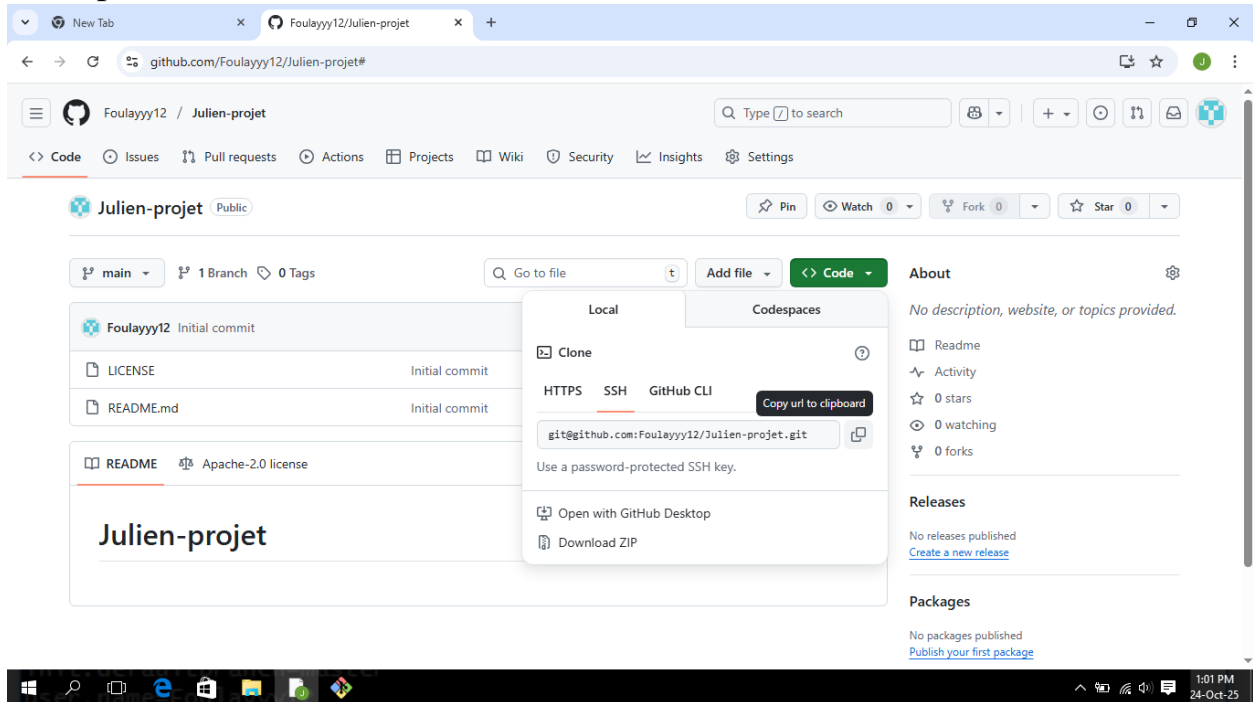


8) Clonage du depot GitHub via SSH

On ouvre GitHub, clique sur code



On copie le code SSH



Après on va sur gitbash et on fait

cd Desktop

git clone [git@github.com :usernane/ mon-projet.git](https://github.com/Foulayyy12/Julien-projet)

ls

cd mon-projet

echo "Systeme d'exploitation" > Module.txt

git add .

git status

git commit -m "Ajout Module"

git branch -M main

git push -u origin main

```
MINGW64/c/Users/marco/Desktop
marco@DESKTOP-7JILCKJ MINGW64 ~
$ cd Desktop

marco@DESKTOP-7JILCKJ MINGW64 ~/Desktop
$ git clone git@github.com:Foulayyy12/Julien-projet.git
Cloning into 'Julien-projet'...
The authenticity of host 'github.com (140.82.112.4)' can't be established.
ED25519 key fingerprint is: SHA256:+D1Y3wvV6TuJ3hbpZisF/zLDA0zPMsvldkr4UvCoQu
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: 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), 4.76 KiB | 68.00 KiB/s, done.

marco@DESKTOP-7JILCKJ MINGW64 ~/Desktop
$ ls
Book1.xlsx
'Calculator Suite.Ink'
'Canva.Ink'
'Charlomagie VICTOR/'
'Clif_Multi-services/'
'CodeBlocks.Ink'
'ETE ius/'
'Examen Excel Mars 2025, Bazile Saglita, Bertrand Raphot.xlsx'
'Finfoj/'
'GoMode.[ED7B470-8E54-465E-825C-99712043E01C]/
'HTML projet/'
'Intra Excel Julien FRANCOIS.xlsx'
'Julien (Person 1) - Chrome.Ink'
'Julien-projet/'
'Menu-setup-abroad-02bf66ec - Shortcut.Ink'
'MinGW Installer.Ink'
'Netflix Installer - Shortcut.Ink'
'New folder/'
'QB13 V.CAN/'
'SamwfoolSetup_v4.9 - Shortcut.Ink'
'VirtualDJ 8.Ink'
'Visual Studio Code.Ink'
code/
desktop.ini
'ius macroeconomie/'
projet-git/

marco@DESKTOP-7JILCKJ MINGW64 ~/Desktop
$
```

```
MINGW64/c/Users/marco/Desktop/Julien-projet
'HTML projet/'
'Intra Excel Julien FRANCOIS.xlsx'
'Julien (Person 1) - Chrome.Ink'
'Julien-projet/'
'Menu-setup-abroad-02bf66ec - Shortcut.Ink'
'MinGW Installer.Ink'
'Netflix Installer - Shortcut.Ink'
'New folder/'
'QB13 V.CAN/'
'SamwfoolSetup_v4.9 - Shortcut.Ink'
'VirtualDJ 8.Ink'
'Visual Studio Code.Ink'
code/
desktop.ini
'ius macroeconomie/'
projet-git/

marco@DESKTOP-7JILCKJ MINGW64 ~/Desktop
$ cd Julien-projet

marco@DESKTOP-7JILCKJ MINGW64 ~/Desktop/Julien-projet (main)
$ echo "Système d'exploitation!" > Module.txt

marco@DESKTOP-7JILCKJ MINGW64 ~/Desktop/Julien-projet (main)
$ git add .
warning: in the working copy of 'Module.txt', LF will be replaced by CRLF the next time Git touches it

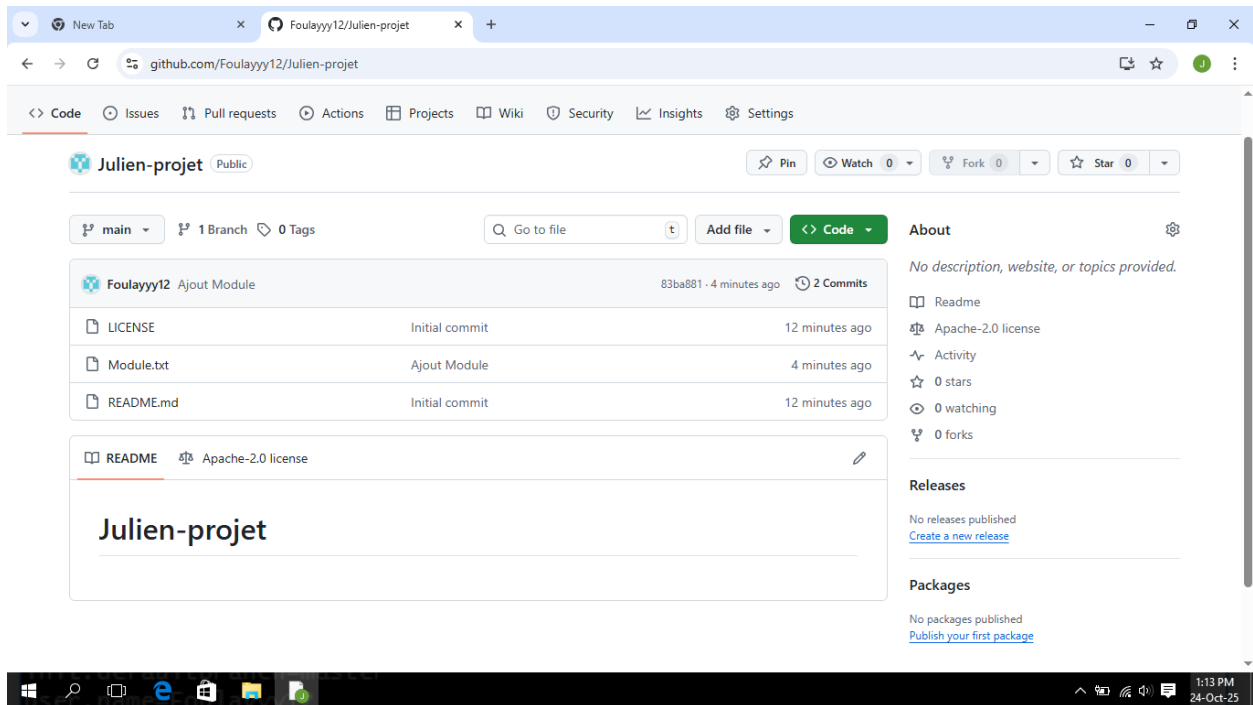
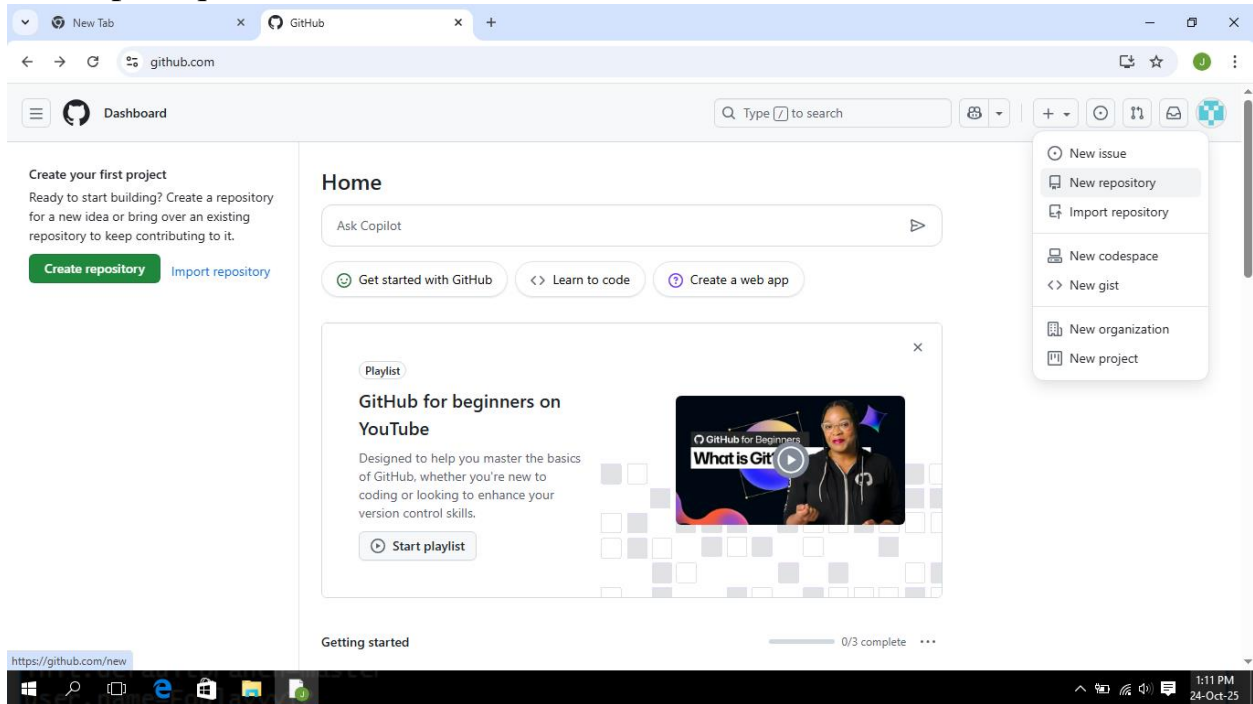
marco@DESKTOP-7JILCKJ MINGW64 ~/Desktop/Julien-projet (main)
$ git commit -m "Ajout Module"
[main 83ba881] Ajout Module
1 file changed, 1 insertion(+)
create mode 100644 Module.txt

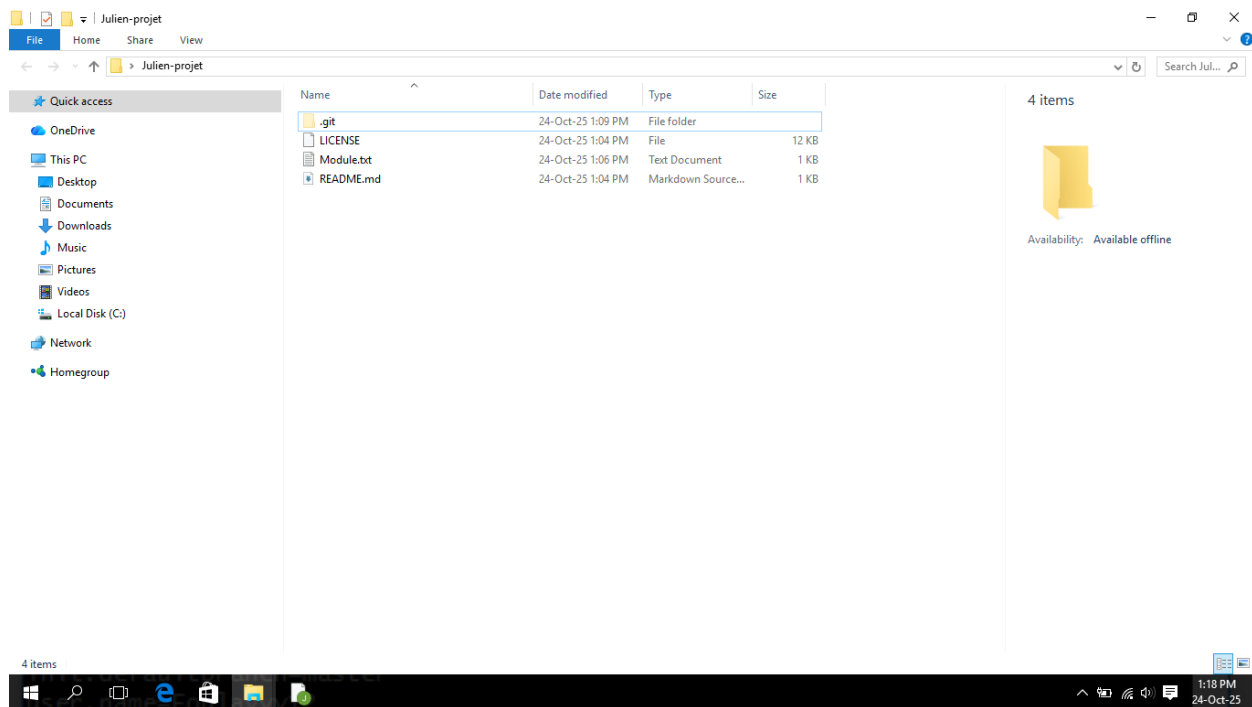
marco@DESKTOP-7JILCKJ MINGW64 ~/Desktop/Julien-projet (main)
$ git branch -M main

marco@DESKTOP-7JILCKJ MINGW64 ~/Desktop/Julien-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), 341 bytes | 170.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Foulayyy12/Julien-projet.git
e11a250..83ba881 main -> main
branch 'main' set up to track 'origin/main'.

marco@DESKTOP-7JILCKJ MINGW64 ~/Desktop/Julien-projet (main)
$
```

Maintenant on va creer un nouveau depot sur GitHub au nom de TD, on va le rendre publique et le cloner





Creation des dossiers dans le dossier TD

```
MINGW64/c/Users/marco/Desktop/TD
marco@DESKTOP-7J1LCK3 MINGW64 ~
$ cd Desktop
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop
$ cd TD
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD (main)
$ mkdir systeme
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD (main)
$ cd systeme
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD/systeme (main)
$ mkdir image
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD/systeme (main)
$ mkdir projet
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD/systeme (main)
$ mkdir presentation
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD/systeme (main)
$ cd ..
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD (main)
$ mkdir reseaul
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD (main)
$ cd reseaul
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD/reseaul (main)
$ mkdir image
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD/reseaul (main)
$ mkdir projet
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD/reseaul (main)
$ mkdir presentation
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD/reseaul (main)
$ cd ..
marco@DESKTOP-7J1LCK3 MINGW64 ~/Desktop/TD (main)
$ |
```

```
MINGW64/c/Users/marco/Desktop/TD
marco@DESKTOP-7JILCK3 MINGW64 ~
$ cd Desktop
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop
$ cd TD
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD (main)
$ mkdir systeme
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD (main)
$ cd systeme
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD/systeme (main)
$ mkdir image
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD/systeme (main)
$ mkdir projet
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD/systeme (main)
$ mkdir presentation
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD/systeme (main)
$ cd ..
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD (main)
$ mkdir reseauI
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD (main)
$ cd reseauI
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD/reseauI (main)
$ mkdir image
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD/reseauI (main)
$ mkdir projet
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD/reseauI (main)
$ mkdir presentation
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD/reseauI (main)
$ cd ..
marco@DESKTOP-7JILCK3 MINGW64 ~/Desktop/TD (main)
$ |
```

Conclusion

Dons mon travail j'ai installé Git, vérifié sa version et l'installation. J'ai créé un compte GitHub, configuré Git, créé des dossiers, créé une clé SSH, fait le clonage de la clé. Pour moi c'était une belle aventure car malgré la difficulté comme créer la clé, mais j'ai réussi mon travail.