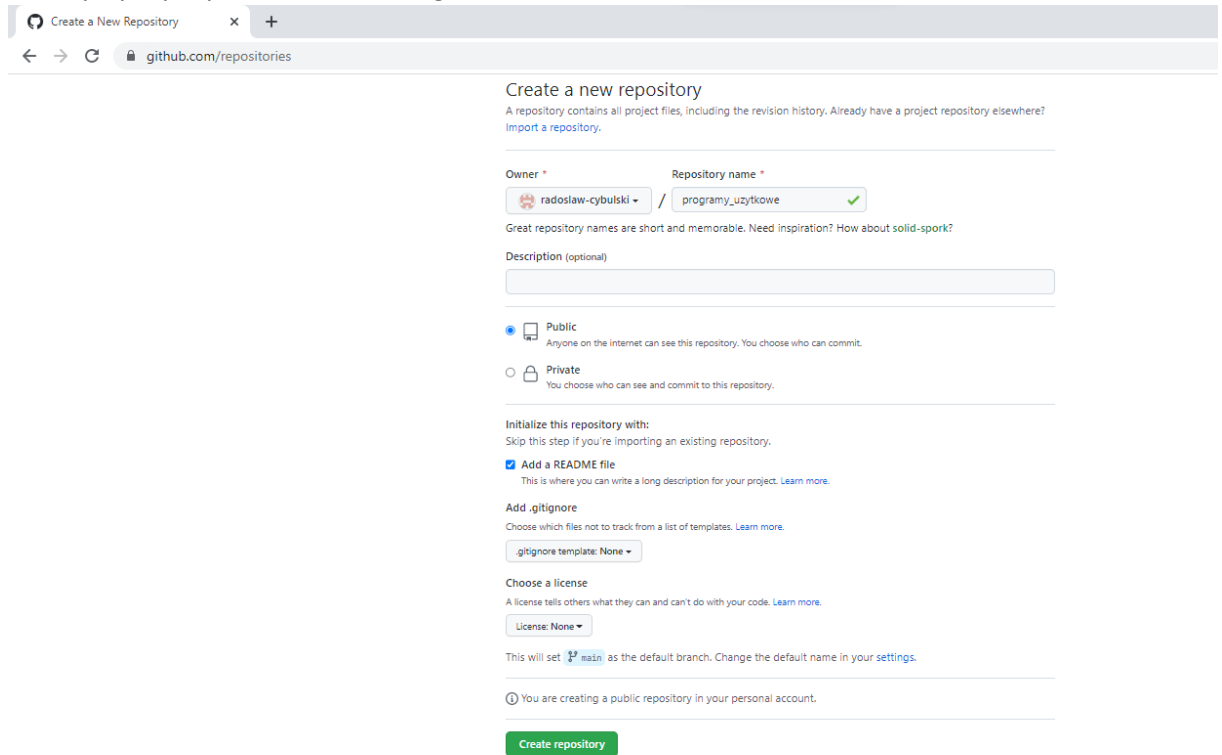


Instrukcja tworzenia repozytorium git krok po kroku.

1. Tworzymy repozytorium na stronie github.com



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * radoslaw-cybulski / Repository name * programy_uzytkowe ✓

Great repository names are short and memorable. Need inspiration? How about solid-spork?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None

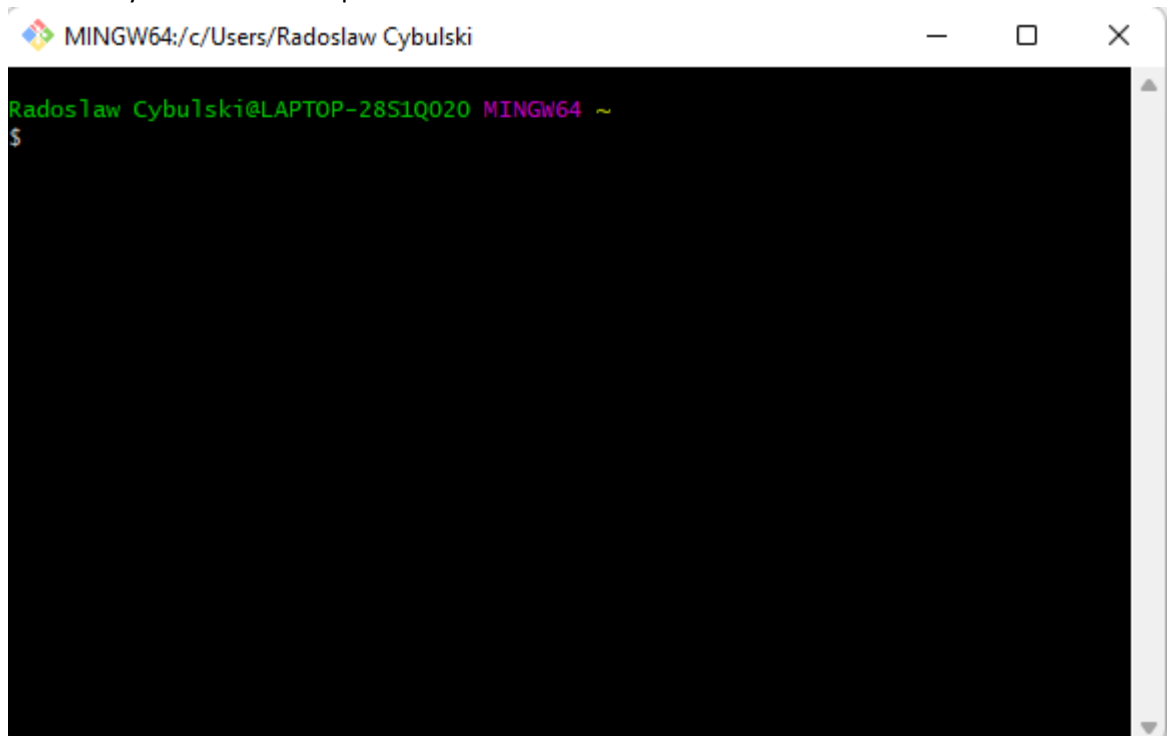
This will set main as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a public repository in your personal account.

[Create repository](#)

W widoku tworzenia repozytorium zaznaczamy opcji 'Public' oraz 'Add a README file', tak jak na zdjęciu wyżej.

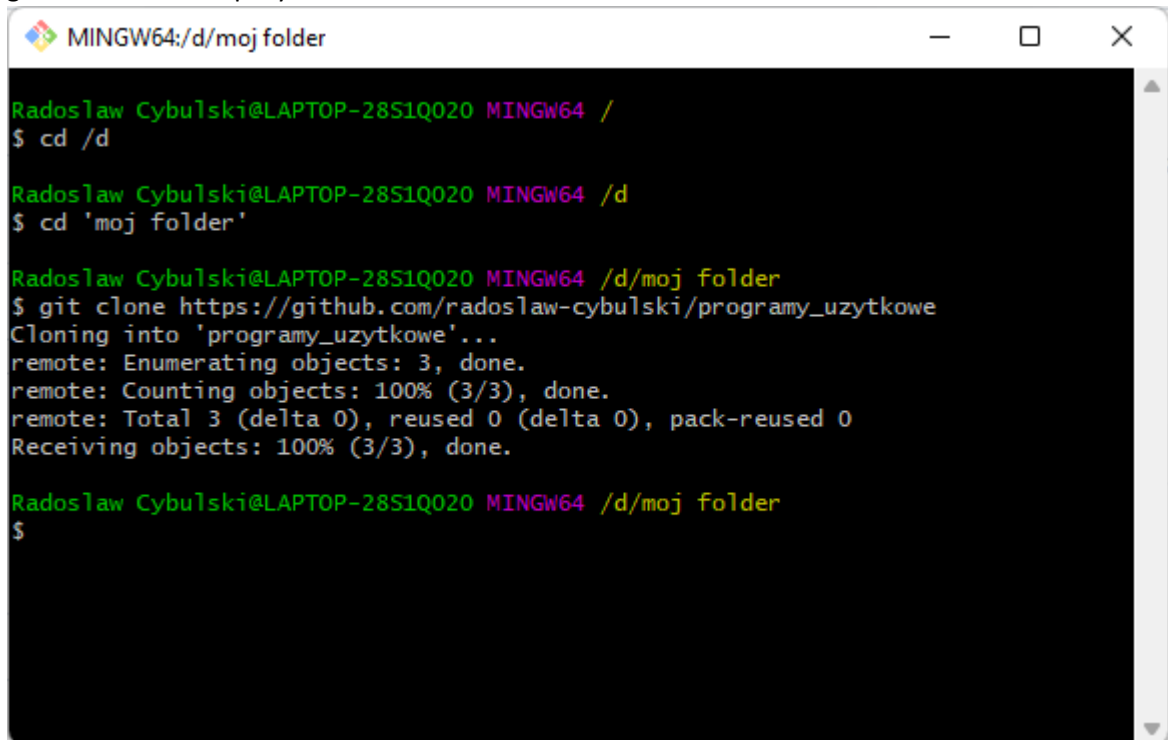
2. Otwieramy Git Bash na komputerze.



3. W konsoli uruchamiamy polecenia ustawiające nazwę użytkownika oraz email (nazwa użytkownika i email muszą być takie same jak na naszym koncie github) poleceniami:
git config --global user.name 'Jan Nowak'
git config --global user.email jannowak@example.com

4. Na dysku D tworzymy własny katalog a następnie przechodzimy do niego poleceniami:
cd /d
cd 'moj folder'

5. Klonujemy repozytorium do naszego folderu poleceniem:
git clone 'link do repozytorium'



```
MINGW64:/d/moj folder

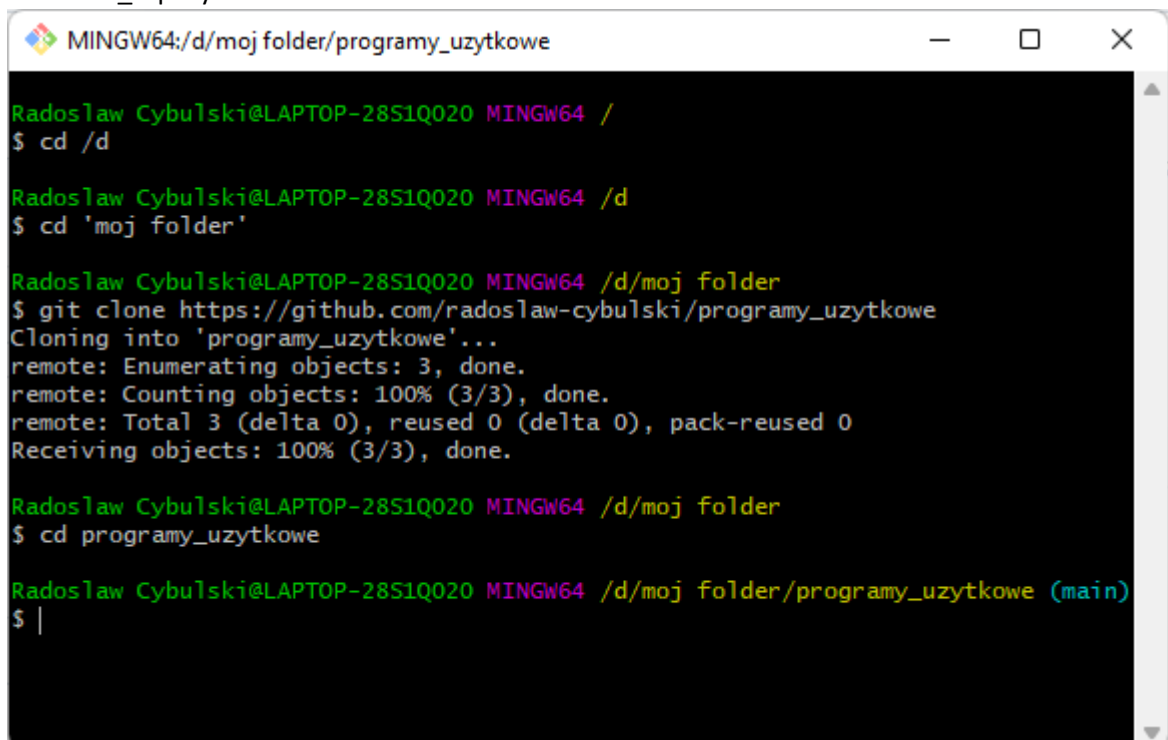
Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /
$ cd /d

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d
$ cd 'moj folder'

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder
$ git clone https://github.com/radoslaw-cybulski/programy_uzytkowe
Cloning into 'programy_uzytkowe'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder
$
```

6. Aby przejść na dysku do sklonowanego repozytorium uruchamiamy poleceniem
cd nazwa_repozytorium



```
MINGW64:/d/moj folder/programy_uzytkowe

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /
$ cd /d

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d
$ cd 'moj folder'

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder
$ git clone https://github.com/radoslaw-cybulski/programy_uzytkowe
Cloning into 'programy_uzytkowe'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder
$ cd programy_uzytkowe

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder/programy_uzytkowe (main)
$ |
```

7. W naszym repozytorium na dysku tworzymy plik tekstowy.
8. Uruchamiamy polecenie git status.

```
MINGW64:/d/moj folder/programy_uzytkowe
Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder
$ git clone https://github.com/radoslaw-cybulski/programy_uzytkowe
Cloning into 'programy_uzytkowe'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder
$ cd programy_uzytkowe

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder/programy_uzytkowe (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        nowy_plik.txt

nothing added to commit but untracked files present (use "git add" to track)

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder/programy_uzytkowe (main)
$
```

9. Uruchamiamy polecenie git add 'nazwa_pliku.rozszerzenie' aby dodać plik do śledzenia przez gita. Następnie jeszcze raz używamy polecenia git status.

```
MINGW64:/d/moj folder/programy_uzytkowe
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        nowy_plik.txt

nothing added to commit but untracked files present (use "git add" to track)

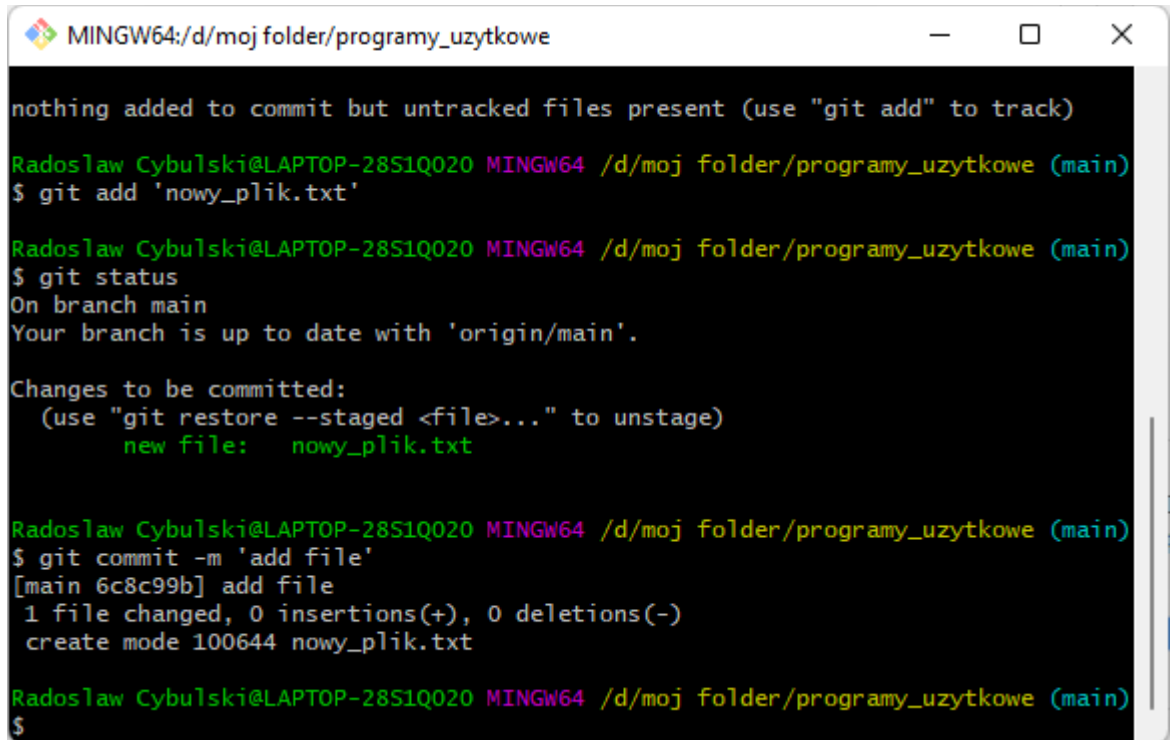
Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder/programy_uzytkowe (main)
$ git add 'nowy_plik.txt'

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder/programy_uzytkowe (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   nowy_plik.txt

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder/programy_uzytkowe (main)
$
```

10. Robimy nowego commita zmian poleceniem
`git commit -m 'opis zmian'`



```
MINGW64:/d/roj folder/programy_uzytkowe

nothing added to commit but untracked files present (use "git add" to track)

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/roj folder/programy_uzytkowe (main)
$ git add 'nowy_plik.txt'

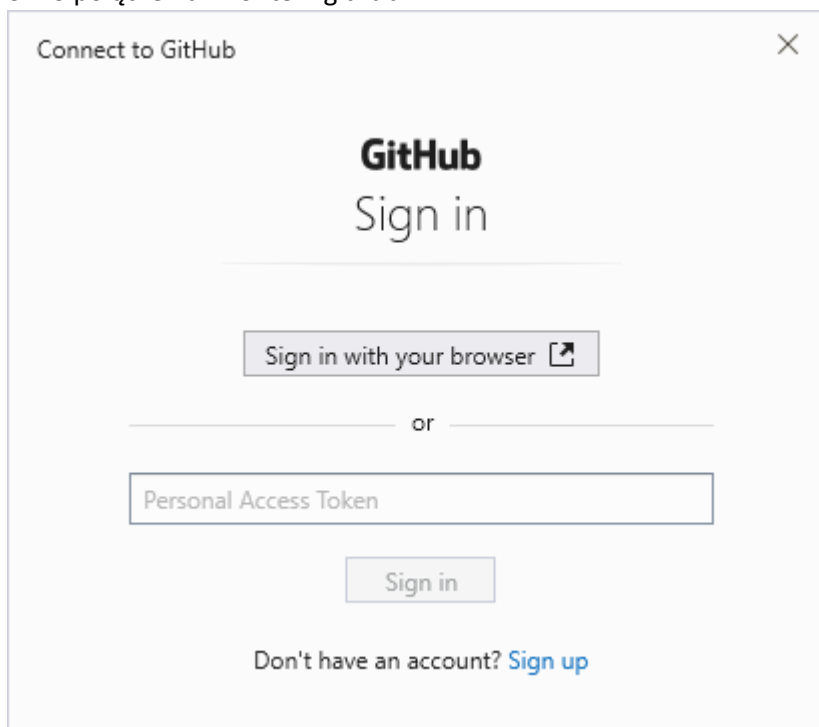
Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/roj folder/programy_uzytkowe (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   nowy_plik.txt

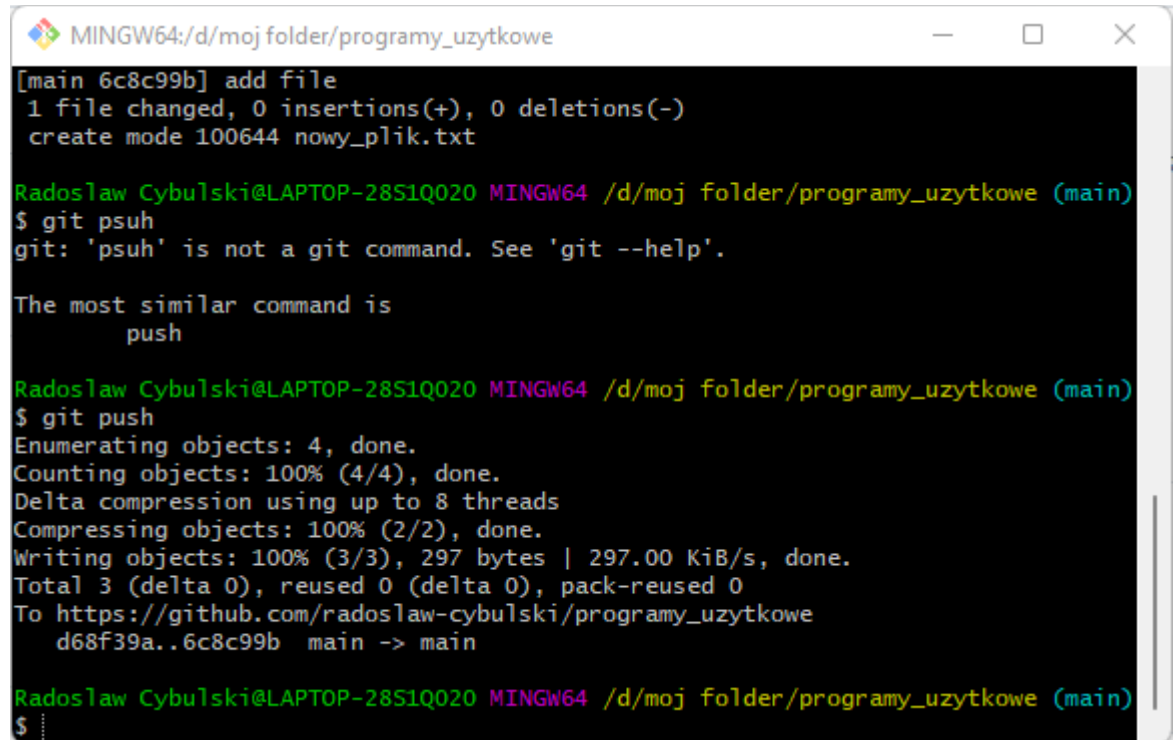
Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/roj folder/programy_uzytkowe (main)
$ git commit -m 'add file'
[main 6c8c99b] add file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 nowy_plik.txt

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/roj folder/programy_uzytkowe (main)
$
```

11. Wypychamy nasze zmiany do github używając polecenia `git push`, następnie otworzy nam się okno połączenia z kontem github.



Klikamy 'Sign in with your browser', następnie logujemy się do swojego konta na github.com. Po uzyskaniu autoryzacji możemy odświeżyć swoje repozytorium. Po niżej zdjęcie po wykonaniu polecenia git push

A screenshot of a terminal window titled 'MINGW64:/d/moj folder/programy_uzytkowe'. The terminal shows the output of a 'git add' command, followed by an attempt to run 'git psuh' which results in an error. Then, the correct command 'git push' is run, showing progress for enumerating, counting, compressing, and writing objects, and finally pushing to the GitHub repository.

```
MINGW64:/d/moj folder/programy_uzytkowe
[main 6c8c99b] add file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 nowy_plik.txt

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder/programy_uzytkowe (main)
$ git psuh
git: 'psuh' is not a git command. See 'git --help'.

The most similar command is
    push

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder/programy_uzytkowe (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 297 bytes | 297.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/radoslaw-cybulski/programy_uzytkowe
d68f39a..6c8c99b  main -> main

Radosław Cybulski@LAPTOP-28S1Q020 MINGW64 /d/moj folder/programy_uzytkowe (main)
$
```