

# Using Git Bash for the Mini-Project

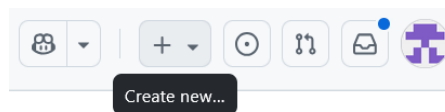
## 1. Git

You must have downloaded Git. If you haven't already done so, go to : <https://git-scm.com/downloads> and select your operating system to get the installation files.

## 2. Github

You will store the project files in a Github directory. 1. You must have an account on <https://github.com>.

2. Then create a new directory :



then "New repository".

3. In the information requested, you only need to add the name. Use the following format : **ProjectDB\_Name1\_Name2** with Name1 and Name2 being the names of the two members of the pair (and with a third if applicable).

4. Then invite the second member of your pair and **your lab instructor** to collaborate :



**Add collaborators to this repository**

Search for people using their GitHub username or email address.

Invite collaborators

5. Initialize the repository with a file (the project topic, or the requested files if you have already started) :

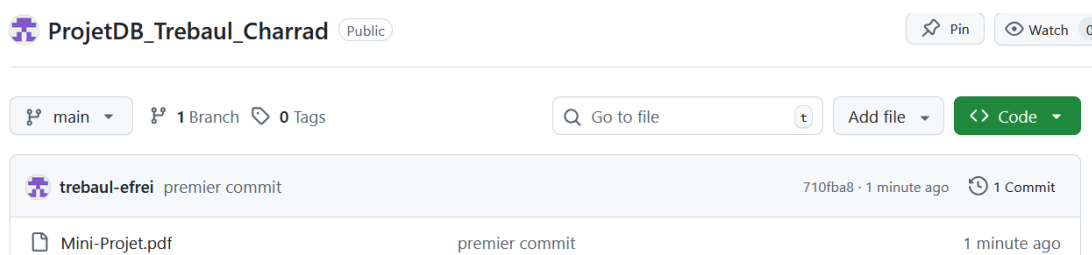
### Quick setup — if you've done this kind of thing before

 Set up in Desktop
 or
 ☐ HTTPS
 ☐ SSH
 [https://github.com/trebaul-efrei/ProjetDB\\_Trebaul\\_Charrad.git](https://github.com/trebaul-efrei/ProjetDB_Trebaul_Charrad.git)

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

by clicking on "Uploading an existing file."

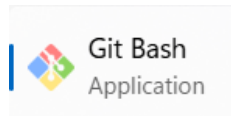
6. At this stage, you have configured your **remote directory**, and you should see something like this :



### 3. Local copy

The directory you have created is stored on a remote server. Each member of your group must have a copy of it on their local machine. To do this, you can use a tool that you downloaded with Git : **Git Bash (terminal)** (it is also possible to use Git GUI (graphical user interface)).

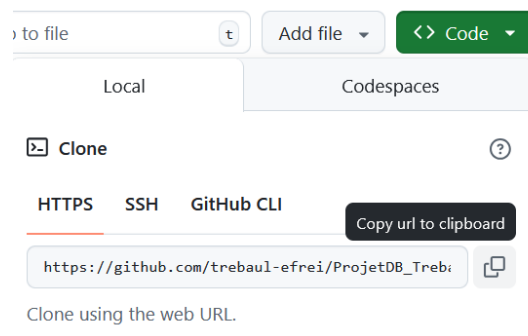
1. Open a git terminal with **Git Bash** :



2. Move to the directory where you want to store the project directory using the `cd` command followed by the directory path, for example :

```
cd "C:\Users\Lena TREBAUL\OneDrive - Efrei\Documents\Cours\BDD\TI503N-2026"
```

3. On GitHub, copy the URL of the remote directory by clicking on "Code"; it should appear below :



4. Create a clone of the remote directory with the command `git clone` followed by the URL, for example :

```
Lena TREBAUL@DESKTOP-4Q1JJNC MINGW64 ~/OneDrive - Efrei/Documents/Cours/BDD/TI503N-2026
$ git clone https://github.com/trebaul-efrei/ProjetDB_Trebaul_Charrad.git
Cloning into 'ProjetDB_Trebaul_Charrad'...
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), 51.70 KiB | 6.46 MiB/s, done.
```

5. Check that the directory has been created with the command `ls` : you should see the name of the directory in the list.
6. Move to the new directory with `cd` , and you should find the file added in the previous section using `ls` :

```
Lena TREBAUL@DESKTOP-4Q1JJNC MINGW64 ~/OneDrive - Efrei/Documents/Cours/BDD/TI503N-2026
$ cd ProjetDB_Trebaul_Charrad/

Lena TREBAUL@DESKTOP-4Q1JJNC MINGW64 ~/OneDrive - Efrei/Documents/Cours/BDD/TI503N-2026/ProjetDB_Trebaul_Charrad (main)
$ ls
Mini-Projet.pdf
```

## 4. Git add, commit, and push

Once you have started the mini-project, when you want to share your work with your group, you must :

1. Add the files that have been modified, whose modifications you want to share, with the `git add` command :

```
$ git add prompt_conception.docx
```

2. Save the changes made to these files with `git commit` followed by a message explaining what you have changed in quotation marks, for example :

```
$ git commit -m "add the prompt for the design part"
[main b7eabfe] ajout du prompt pour la partie conception
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 prompt_conception.docx
```

3. Send the changes to the remote repository with `git push` :

```
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 12.27 KiB | 12.27 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/trebaul-efrei/ProjetDB_Trebaul_Charrad.git
710fba8..b7eabfe main -> main
```

They should appear in your GitHub repository.

## 5. Git pull

As soon as you resume your work, check that no other member of your group has made any changes to the repository.

To do this, run `git pull` :

```
$ git pull
Already up to date.
```

“Already up to date” indicates that there have been no changes. If this is the case, the files are retrieved to your local machine from the GitHub repository, as shown here :

```
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 3.73 KiB | 212.00 KiB/s, done.
From https://github.com/trebaul-efrei/ProjetDB_Trebaul_Charrad
   b7eabfe..4d72fae main       -> origin/main
   * [new branch]      master   -> origin/master
Updating b7eabfe..4d72fae
Fast-forward
 0_mcd.loo | Bin 0 -> 82852 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 0_mcd.loo
```

### Reminder of best practices

- Divide tasks among group members so that you are not working on the same files at the same time.
- As soon as you resume work on your file, check for changes in the GitHub directory with 'git pull'.
- When you have finished working on a section, make a commit ('git add' then 'git commit') with a message indicating what you have completed,
- At the end of a work session (or before), do a 'git push',
- Tell your group members that you have made a change so that they can do a 'git pull' on their local directory.