Git & GitHub

What is Git?

➔ Version control system for tracking changes in computer files.
- ✓ Distributed/decentralized version control: many developers can work on a single project without having to be on the same network.
- ✓ Coordinates work between multiple developers.
- ✓ Who made what changes and when?
- ✓ Revert back at any time
- ✓ You can push your local repo to the remote one
- ✓ Local repo: you do not need an internet connection, but if you want to push it to the remote repo you will need an internet connection.

What is GitHub?

➔ GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

Concepts of Git:

- ➢ Keeps track of code history
- ➢ Take “snapshots” of your files.
- ➢ You decide when to take a snapshot by making a “commit”
- ➢ You can visit any snapshot at any time
- ➢ You can stage files before committing.
- ➢ Once you make a commit to the remote repos, other developers can pull that information onto their machines.

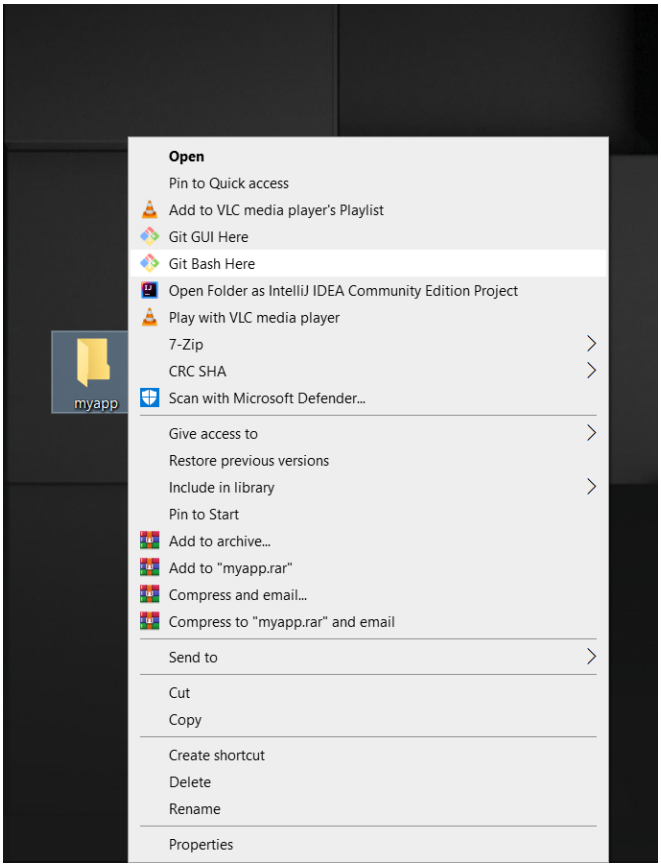
Basic commands:

- git init -> Initialize Local Git Repository
- git add <file> -> add file(s) to index
- git status -> Check status of working tree
- git commit -> Commit changes in index
- git push -> push to remote repository
- git pull -> pull latest changes from remote repository
- git clone -> copy remote repo into a new directory (folder) in your local machine

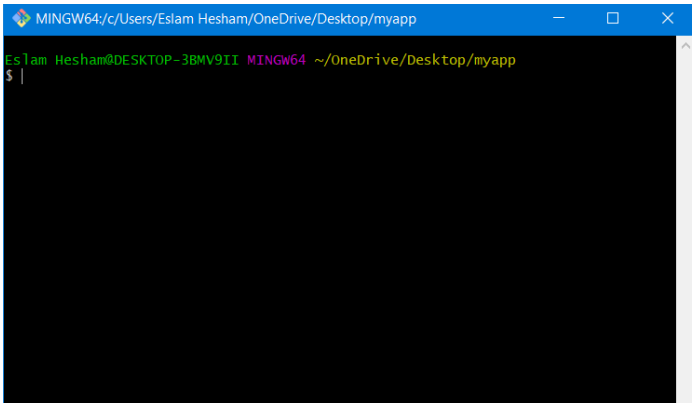
So, let's get started!!

-First, we will create a new folder called myapp.

-Open that folder using git bash.



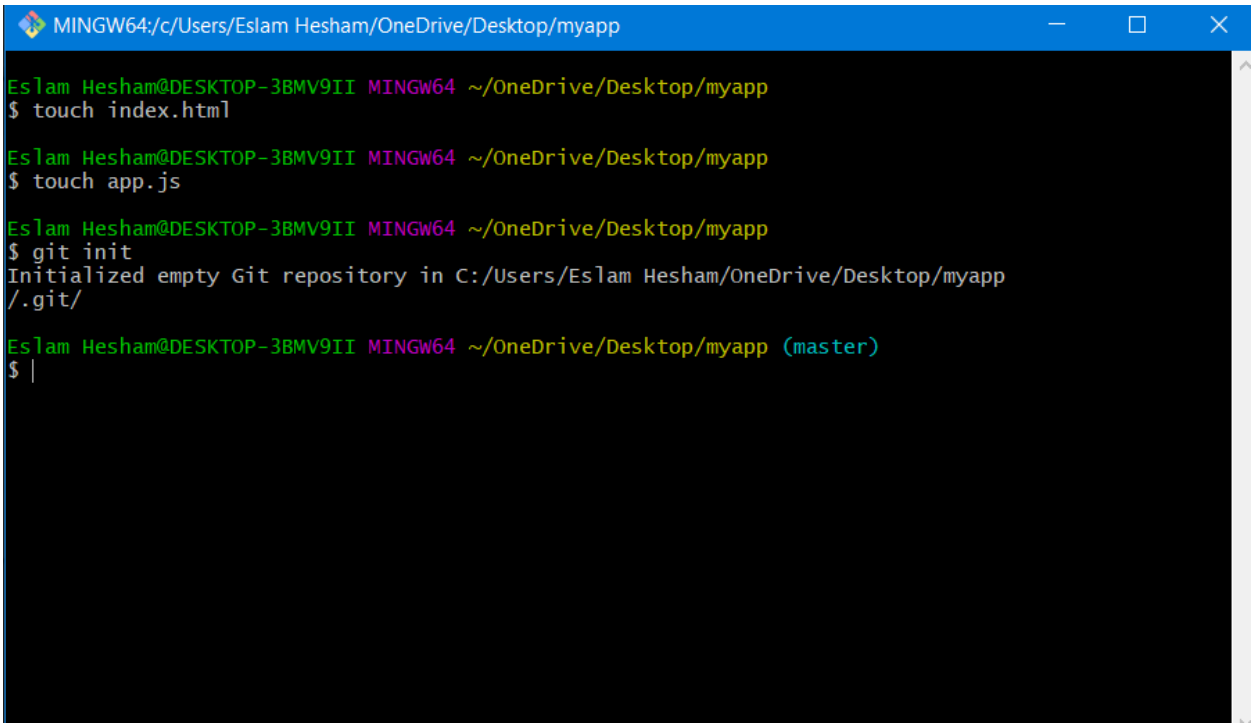
-A git bash window will be opened for myapp folder



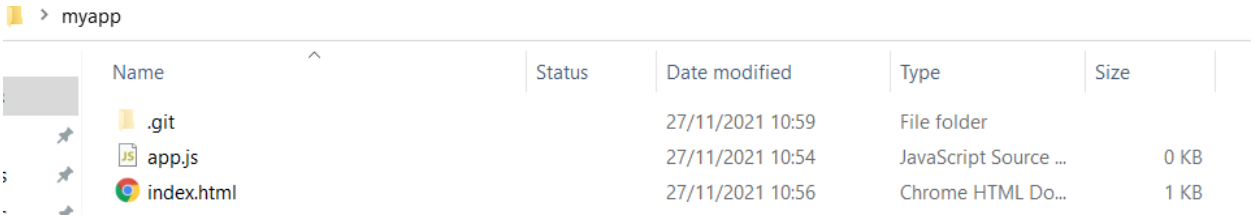
-So, lets create a couple of files that our project would have by either command or file manager.

-we want to initialize this folder as a git repository by using command:

-it will create. git folder in your directory.



-it will create. git folder in your directory.



-We want to add name and email to git, so we use config command:

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git config --global user.name 'Eslam Hesham'

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git config --global user.email 'eslam.hesham.a90@gmail.com'
```

-Let’s add our index.html file to our git repository

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git add index.html
```

-We can check that the file is added to the staging area by using git status command:

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html

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

-If we want to remove index.html file from the staging area, we can use git rm command:

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git rm --cached index.html
rm 'index.html'
```

-If we want to check that index.html file is removed from staging area, we can use git status command:

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        app.js
        index.html

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

-Creating multiple html files for testing:

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ touch about.html

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ touch contact.html

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git rm --cached index.html
rm 'index.html'

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        about.html
        app.js
        contact.html
        index.html

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

-Different ways to add files, for example: **git add *.html** -> it will add any html files to the staging area.

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git add *.html
```

-If we check for the status now, all html files were added successfully.

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   about.html
        new file:   contact.html
        new file:   index.html

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

-If you want to add all files regardless their extensions, we can use git . command

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        about.html
        app.js
        contact.html
        index.html

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

-Adding all files

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git add .
```

-Result after adding files

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   about.html
        new file:   app.js
        new file:   contact.html
        new file:   index.html
```

-What if we add some changes to our files while it is in the staging area?

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   about.html
        new file:   app.js
        new file:   contact.html
        new file:   index.html

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html
```

➔ Since we added some change to our file, so it is telling us to add it to our repo by using git add command.

-We want to capture a snapshot of the project's currently staged changes, we can use git commit

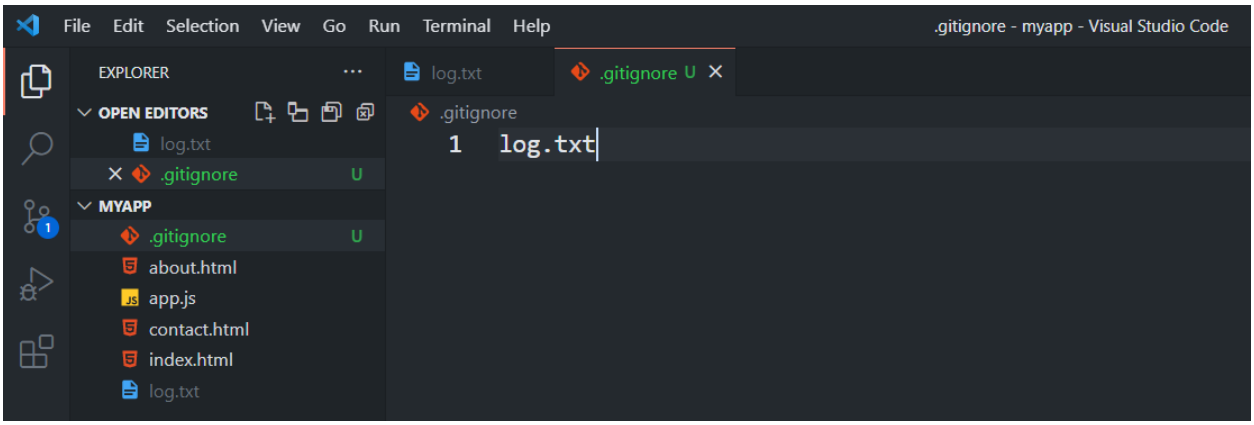
```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git commit -m "Changes made to index, app files"
[master ed3553b] Changes made to index, app files
 2 files changed, 2 insertions(+), 1 deletion(-)
```

-What if we want to ignore some files or folders that we do not want to include them in our repository, by creating. gitignore file.

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ touch .gitignore
```

-We also want to create a file that we don't want to include, for example log.txt

So, we would add the filename inside that. gitignore file that we created.



-Let's create 2 directories, each one with its own files.

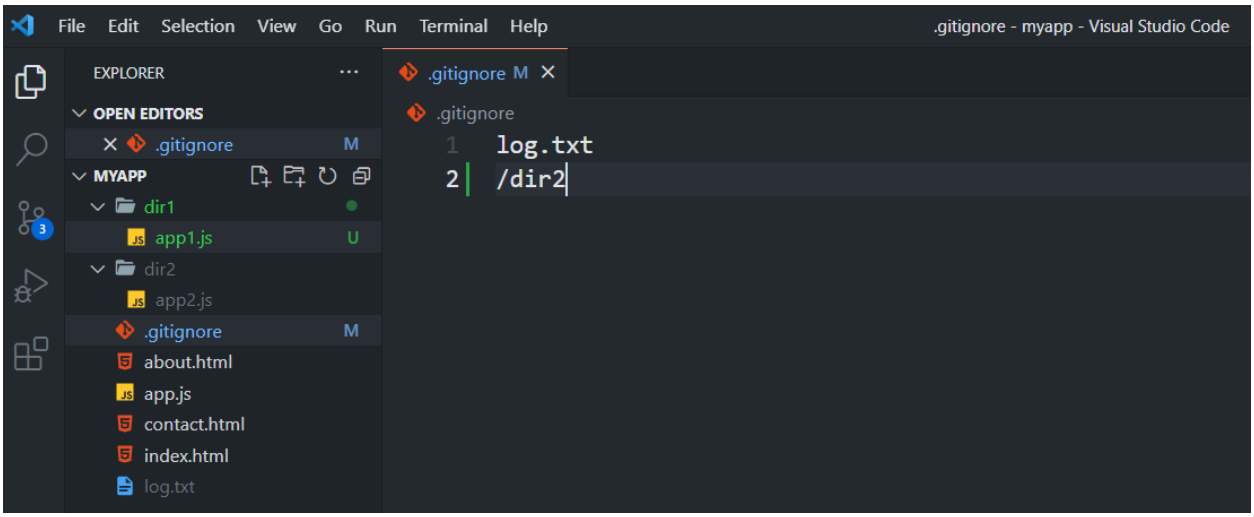
-First directory is called: **dir1**, inside dir1: create a new file called **app1.js**

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ mkdir dir1

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ cd dir1/

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp/dir1 (master)
$ touch app1.js

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp/dir1 (master)
$ cd ..

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
```

-Second directory is called: **dir2**, inside dir2: create a new file called **app2.js**

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ mkdir dir2

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ cd dir2

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp/dir2 (master)
$ touch app2.js

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp/dir2 (master)
$ cd ..
```

-We want to ignore the directory (folder) called: **dir2**



-Let's check that **dir2** is not added within staging area to make sure that it is ignored.

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git add .

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore
        new file:   dir1/app1.js
```

Branches:

Main branch: master

-To create a branch, we can use git branch branchName

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git branch login
```

-To switch to the other branch, we can use git checkout branchName

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git branch login

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git checkout login
Switched to branch 'login'
M       .gitignore

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (login)
$
```

-So, inside this login branch, let's create a new file called login.html

Add that file, also committing by using git add., git commit -m "Comment".

-If we switch back to the main branch, all the files that have been made and the edited changes is going to disappear, because that was in the login branch.

-Now if we want to merge that, if we finish the functionality and we are ready to merge, we can say while we are in the master branch: git merge branchName

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git merge login
Updating 3f86b6b..5c7cf8a
Fast-forward
 .gitignore | 3 ++-
 index.html | 2 +-
 login.html | 1 +
 3 files changed, 4 insertions(+), 2 deletions(-)
 create mode 100644 login.html
```

-So, now even we are in the master branch, we can see all the changes made in the other branch.

-Now we want to link this git local project to a git remote repo (GitHub).

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git remote add origin git@github.com:EslamHesham404/sampleApp.git

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git remote
origin
```

-Origin: name of the remote repo.

-If we want to list remote repos, we can use **git remote** command.

-We want to push that local folder to the remote repository, which is called origin, we can use git push command:

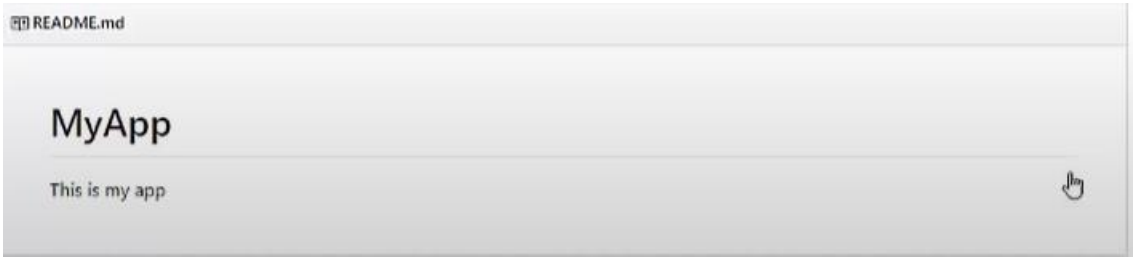
```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git push -u origin master
```

A pop up screen will be shown asking for the username and password to prove your authenticity in order to have the access rights to push to that remote repo.

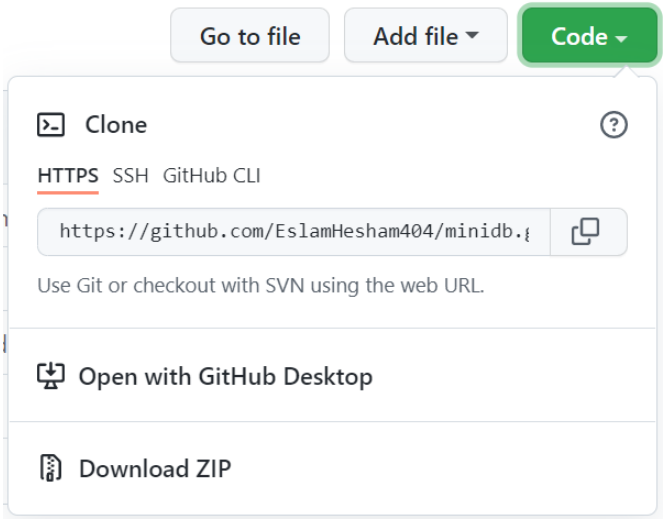
-Let’s add our README.md file to help people understand your project.

```
Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ touch README.md

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git add .

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git commit -m "Added readme"
[master ec54118] Added readme
 1 file changed, 2 insertions(+)
 create mode 100644 README.md

Eslam Hesham@DESKTOP-3BMV9II MINGW64 ~/OneDrive/Desktop/myapp (master)
$ git push
```

-README.md file is displayed.

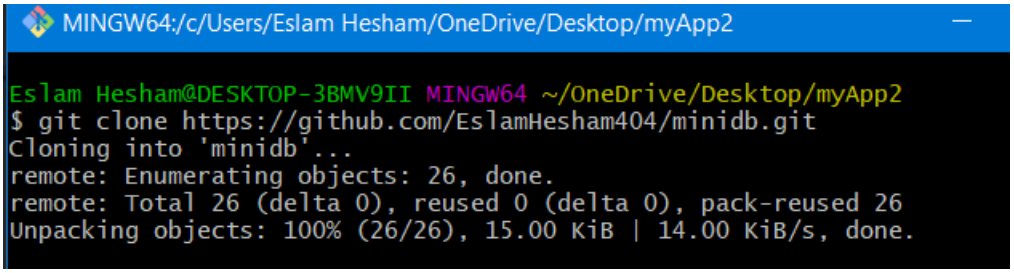


-What if we want to clone a project from GitHub (remote repo) into our local machine?

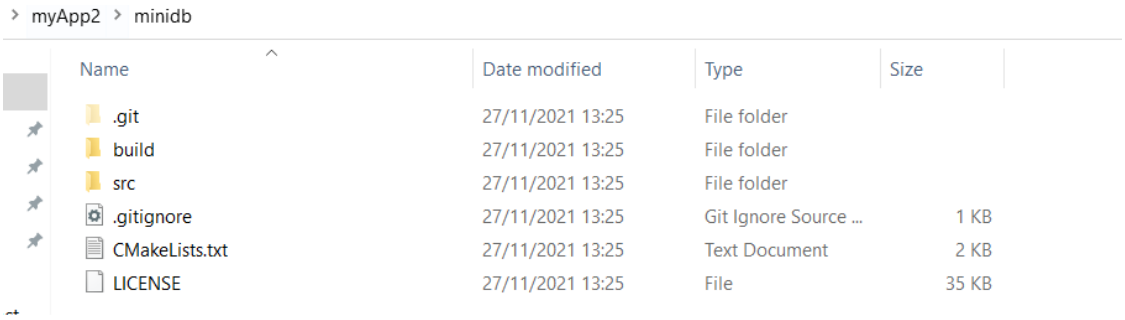


-Project files on GitHub

-So, lets create a new folder called myApp2 and open this folder with git bash, we can use git clone command:



-And now you can see we have the entire application here.



-What if we want to update the local version of a repository from a remote, we can use git pull command: