Repository Synchronizer

Submitted By:

AP18110010151	-	P.Gowtham
AP18110010152	-	K.Rushya Reddy
AP18110010153	-	S.Thanuja Pavani
AP18110010154	-	k.Sravya
AP18110010155	-	V.Sarayu
AP18110010156	-	P.Sampath Ranga Sai

CSE-C

Problem Statement:

So far we have been doing all our tasks locally.This means that all the changes that are commited to the local machine only.So, other developers have no idea nor access to our work.Other developers in the same team should be able to view and contribute to the project at the same time without overwriting eachother’s work.To make it possible we can use Repository Synchronization where we use remote repositories.

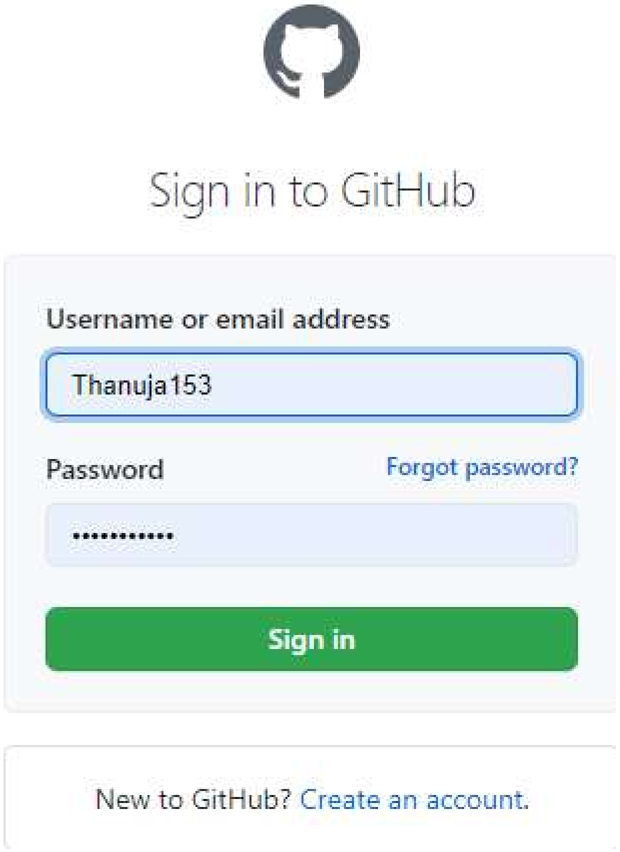
Introduction:

Repository is a storage location for storing files , documents , programs ,projects safely etc**.** Process Synchronization means sharing system resources by processes in a such a way that, Concurrent access to shared data is handled thereby minimizing the chance of inconsistent data. Here We are going to do Repository Synchronization. Repository Synchronization is a process of ensuring the files,documents etc in repository are updated via certain rules(like no overwriting).Here we take github as our server and users as clients.

Steps to show how Repository Synchronization takes place in Client Server application(github):

1. Create github account

Here github is the Server and Client is the User.



Sign in to GitHub

Username or email address

Thanuja153

Password

Forgot password?

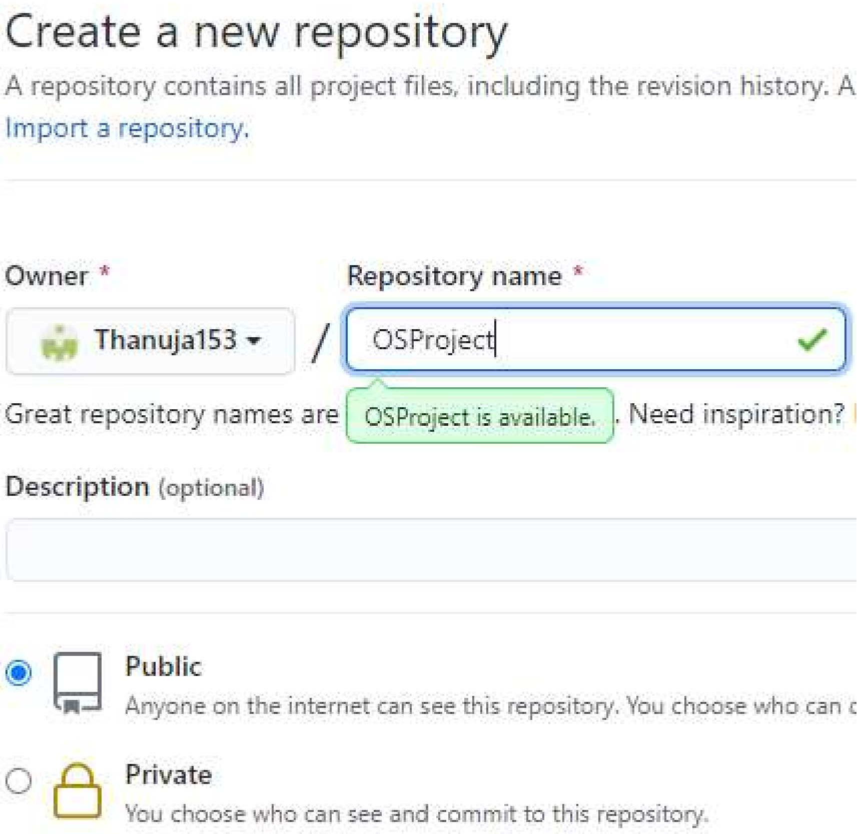
•••••••••••

Sign in

New to GitHub? Create an account.

2. Create a new repository in github.

Repository: it is a place to store important files, documents etc.



Create a new repository

A repository contains all project files, including the revision history. A

Import a repository.

Owner * Thanuja153 / Repository name * OSProject ✓

Great repository names are OSProject is available. Need inspiration?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can c

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

3. Create a Local Folder on your desktop . Insert files into your local folder to insert into your repository.

4.Open gitbash/terminal and go to your local folder directory by using “change directory” command

(cd directory_name)

Steps to Follow:

Step1: cd desktop

Step2: cd Osproject

Step3: git init

The git init command is the first command that we will run on Git. The git init command is used to create a new blank repository. It is used to make an existing project as a Git project. The git init command creates a .git subdirectory in the current working directory. This newly created subdirectory contains all of the necessary metadata. These metadata can be categorized into objects, refs, and temp files. It also initializes a HEAD pointer for the master branch of the repository.

Step5:git status

The list of all untracked files are displayed by the git status command.

Step4: git add filename

The git add command is used to add file contents to the Index also known as Staging Area. When we add a file in git , it will take place in the staging area.The staging **area** is a working area where files are not handled by **git**. These files are also referred to as "untracked files". **Staging area** files are the files that are going to be a part of the next commit, which make **git** to know what changes in the file (previous) that are going to occur for the next commit.We can add single or multiple files at once in the staging area.

To add multiple files:

We use **git add .**

Step5: git Commit –m “some message”

It is used to record/save the changes in the repository. After adding files we need to use this command. Every commit contains the index data and the commit message.

Step6:
git remote add origin https://github.com/Thanuja153/OSProject.git

remote repository: is a common repository that all team members use to exchange their changes.

To **add** a new **remote(repository)**, we use the **git remote add** command on the terminal, in the directory your repository is stored at.We have to store our remote repository url into remote variable origin.

Step7: git branch –M main

It is used to rename old branch name to new branch name.Here we are renaming master branch as main.

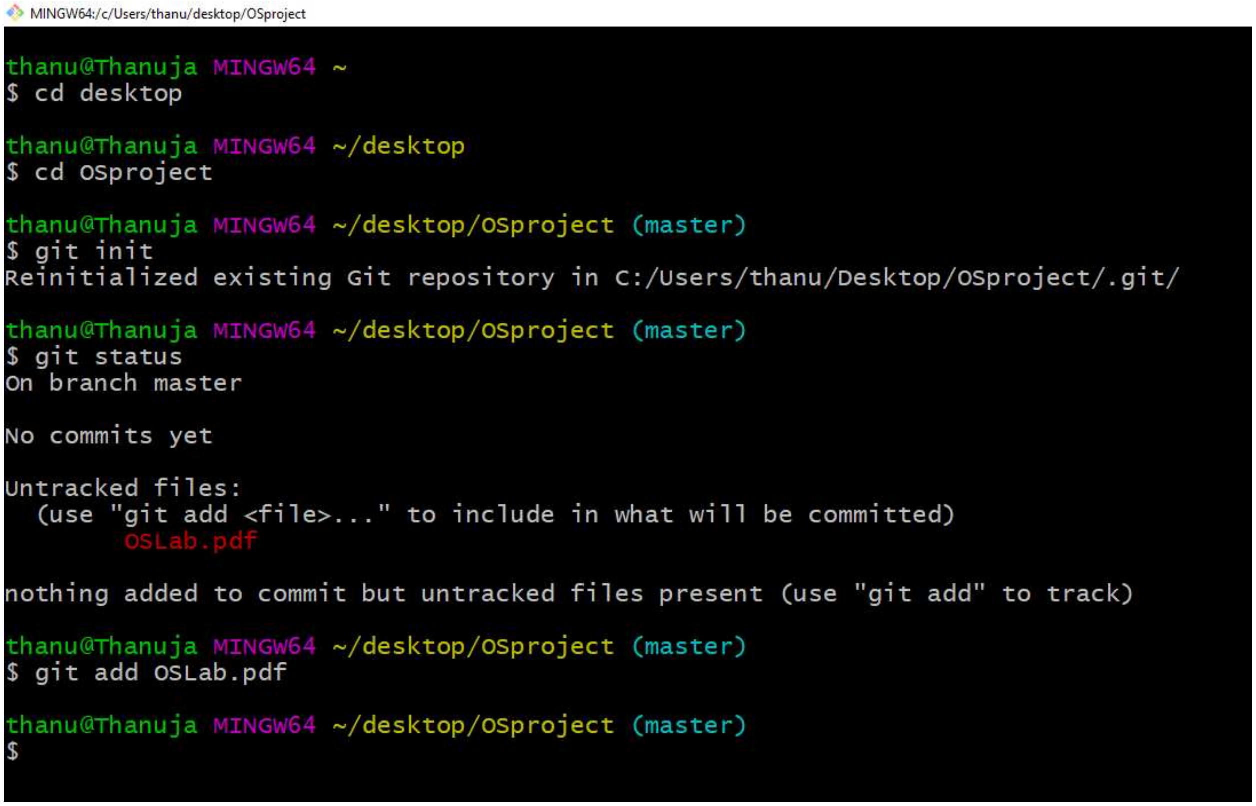
Step8:git push –u origin main

This command helps to create a new tracking connection with the new origin/main branch.

-**u** helps us to capture the remote branch **we** intend to track.

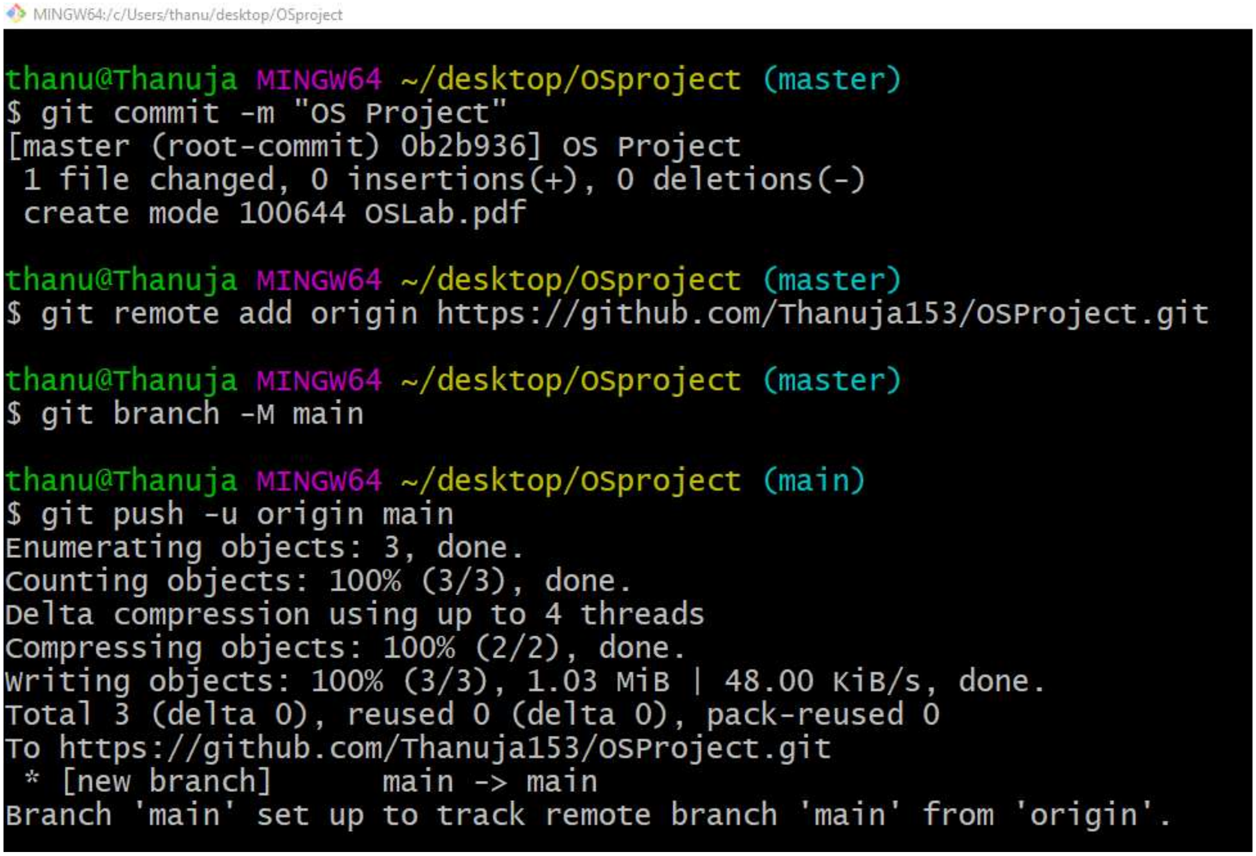
After commit push files to github repository.

Outputs:

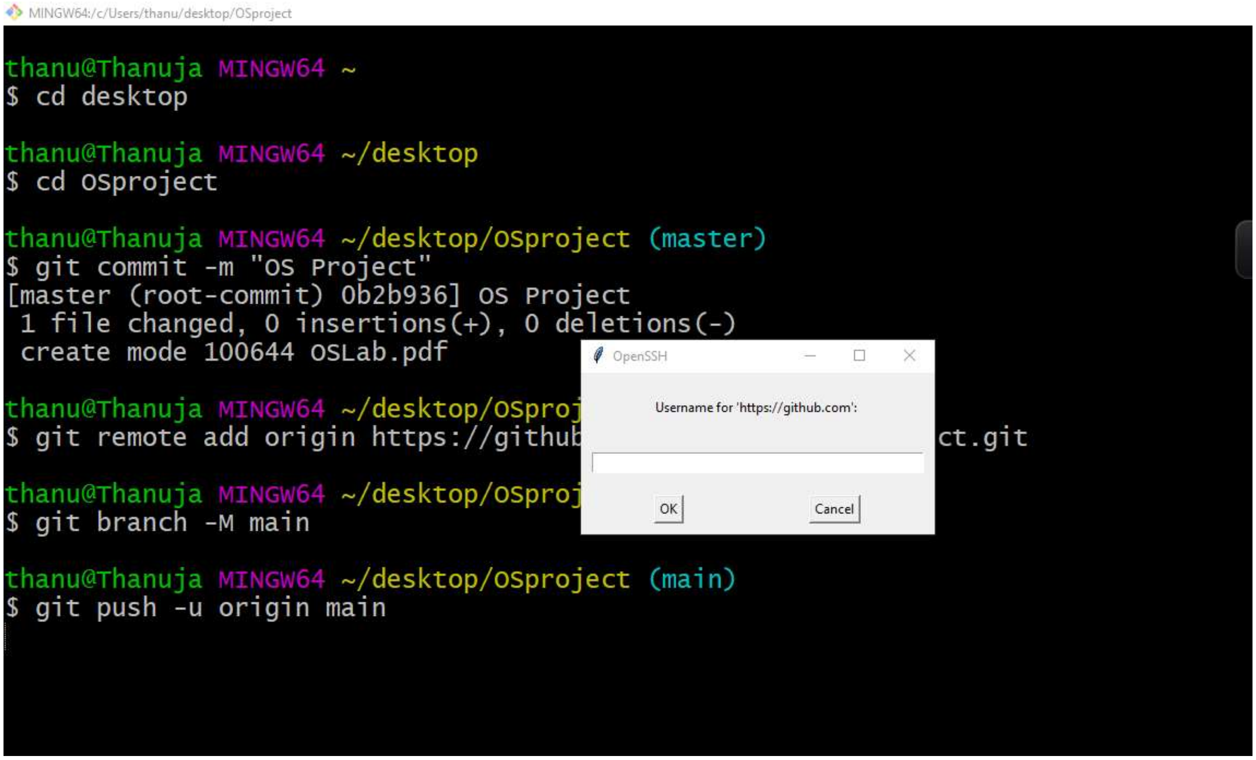


MINGW64:/c/Users/thanu/desktop/OSproject

```
thanu@Thanuja MINGW64 ~
$ cd desktop

thanu@Thanuja MINGW64 ~/desktop
$ cd OSproject

thanu@Thanuja MINGW64 ~/desktop/OSproject (master)
$ git init
Reinitialized existing Git repository in C:/Users/thanu/Desktop/OSproject/.git/

thanu@Thanuja MINGW64 ~/desktop/OSproject (master)
$ git status
On branch master

No commits yet

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

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

thanu@Thanuja MINGW64 ~/desktop/OSproject (master)
$ git add OSLab.pdf

thanu@Thanuja MINGW64 ~/desktop/OSproject (master)
$
```

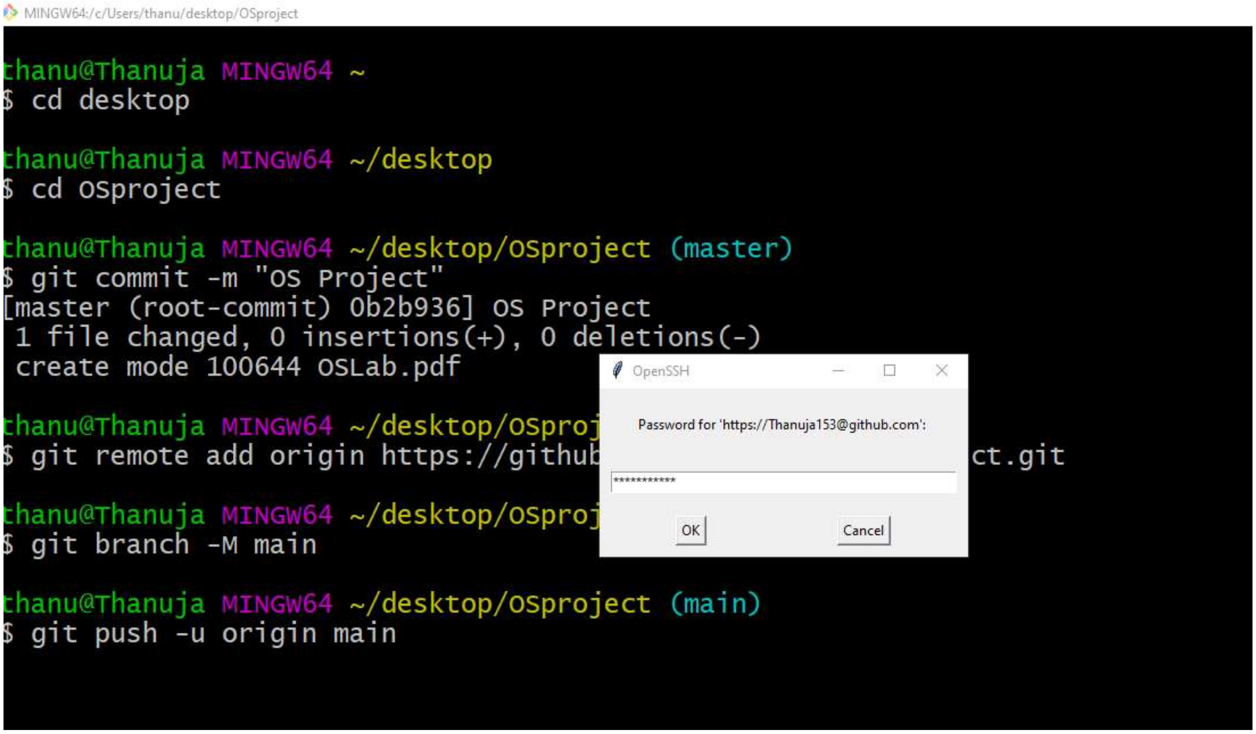


```
thanu@Thanuja MINGW64 ~/desktop/OSproject (master)
$ git commit -m "OS Project"
[master (root-commit) 0b2b936] OS Project
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 OSLab.pdf

thanu@Thanuja MINGW64 ~/desktop/OSproject (master)
$ git remote add origin https://github.com/Thanuja153/OSProject.git

thanu@Thanuja MINGW64 ~/desktop/OSproject (master)
$ git branch -M main

thanu@Thanuja MINGW64 ~/desktop/OSproject (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.03 MiB | 48.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Thanuja153/OSProject.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```



```
thanu@Thanuja MINGW64 ~
$ cd desktop

thanu@Thanuja MINGW64 ~/desktop
$ cd OSproject

thanu@Thanuja MINGW64 ~/desktop/OSproject (master)
$ git commit -m "OS Project"
[master (root-commit) 0b2b936] OS Project
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 OSLab.pdf

thanu@Thanuja MINGW64 ~/desktop/OSproj
$ git remote add origin https://github                    ct.git

thanu@Thanuja MINGW64 ~/desktop/OSproj
$ git branch -M main

thanu@Thanuja MINGW64 ~/desktop/OSproject (main)
$ git push -u origin main
```

Username for 'https://github.com':


MINGW64:/c/Users/thanu/desktop/OSproject

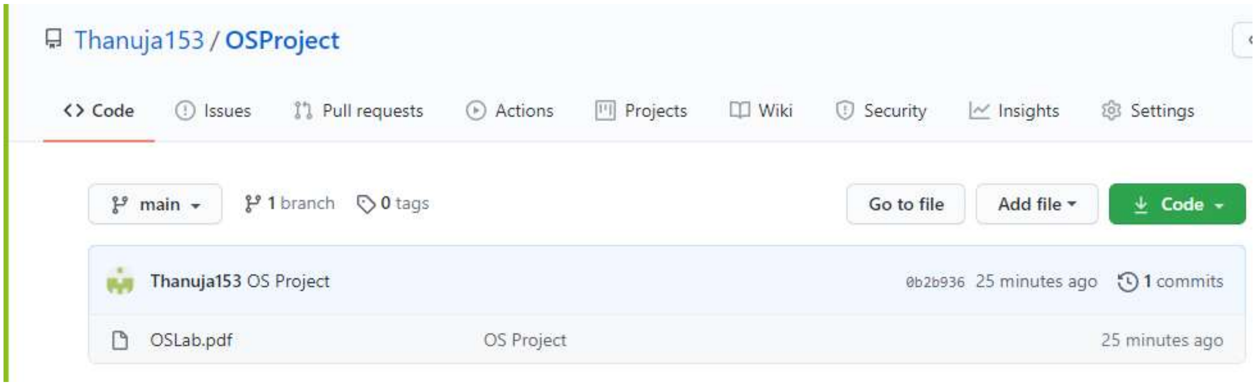
```
thanu@Thanuja MINGW64 ~
$ cd desktop

thanu@Thanuja MINGW64 ~/desktop
$ cd OSproject

thanu@Thanuja MINGW64 ~/desktop/OSproject (master)
$ git commit -m "OS Project"
[master (root-commit) 0b2b936] OS Project
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 OSLab.pdf

thanu@Thanuja MINGW64 ~/desktop/OSproj
$ git remote add origin https://github                          ct.git

thanu@Thanuja MINGW64 ~/desktop/OSproj
$ git branch -M main

thanu@Thanuja MINGW64 ~/desktop/OSproject (main)
$ git push -u origin main
```

OpenSSH

Password for 'https://Thanuja153@github.com':

****

When we use push command it asks for our github account.


Thanuja153 / OSProject

1 branch 0 tags

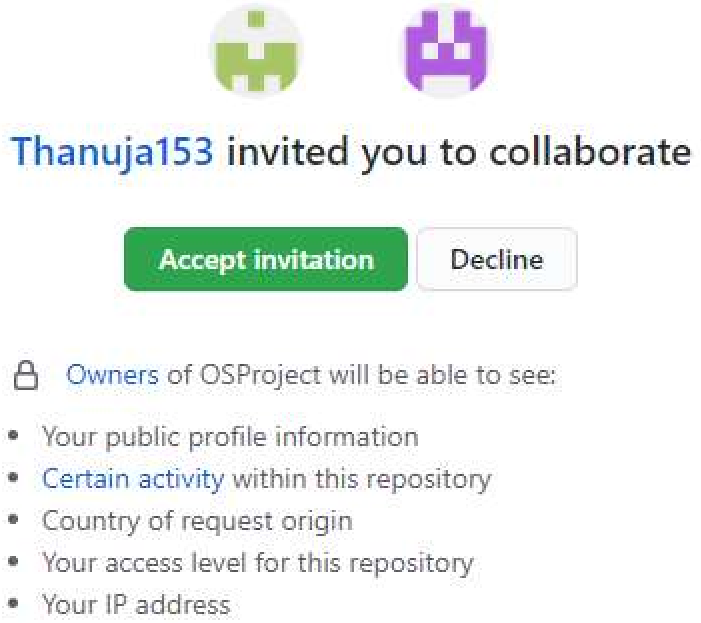
Thanuja153 OS Project 0b2b936 25 minutes ago 1 commits

OSLab.pdf OS Project 25 minutes ago

In the above output we can see that that files are inserted.

Steps to access others repository/masters repository(Cloning):

1.First collaborate with other users/developers.



Thanuja153 invited you to collaborate

Owners of OSProject will be able to see:

- Your public profile information
- Certain activity within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

2. After contributor accepts request from other users,users can clone and modify the master project.

Steps to follow:

Step 1:git clone url(repository branch)

The **git clone** command copies an existing **Git** repository.**git clone** is primarily used to point to an existing repo and make a **clone** or copy of that repo at in a new directory, at another location.

Step 2:git init

Intializiating new blank repository.

Step 3:git remote –verbose

From which repository files are to be fetched or to be added.Showing remote repositories.

Step 4:git add .

Adding files.

Step 5:git status

To check how many new files to be committed.

Step 6: git commit –m “OS”

To save the changes.

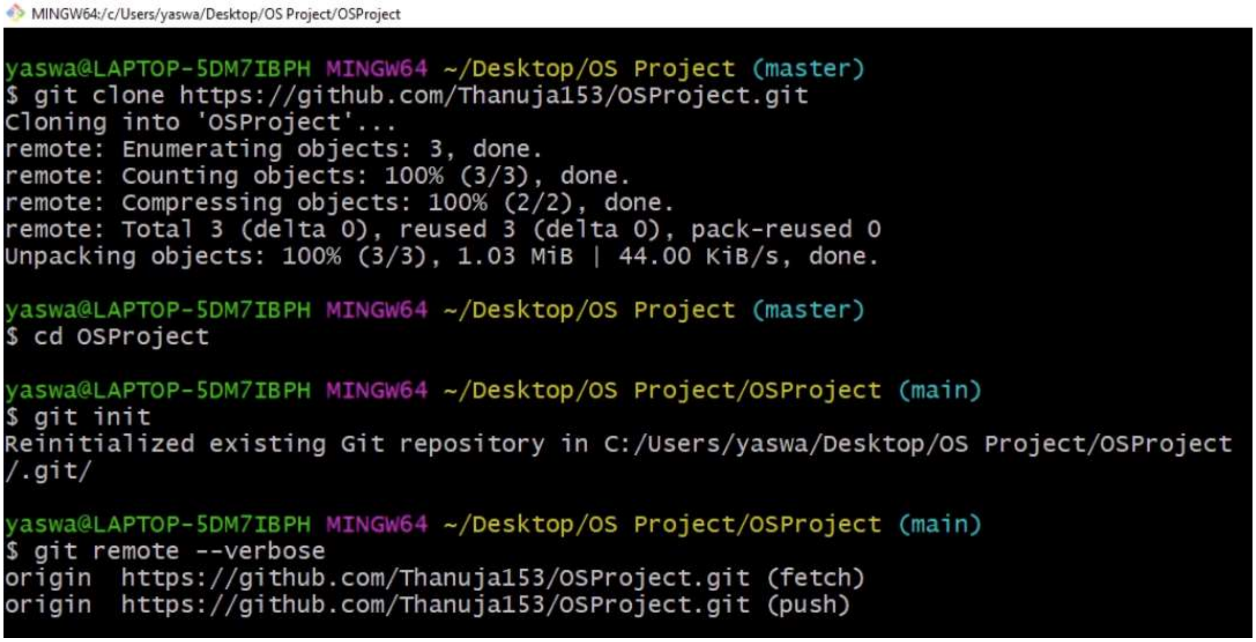
Step 7: git status

Step 8 : git branch –M main

On which repository sub file we are working on.

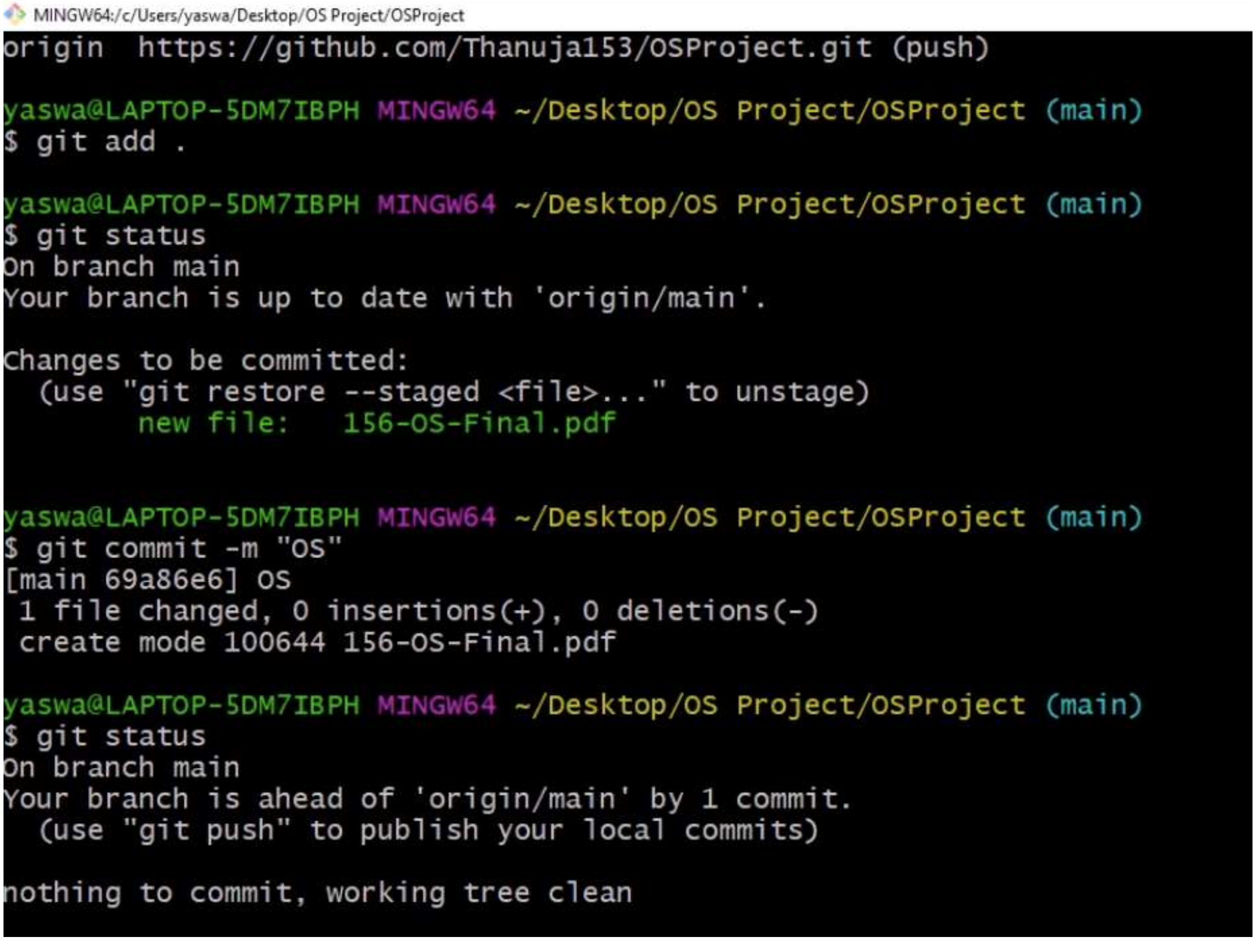
Step 9 : git push –u origin main

Pushing new changes to the repository



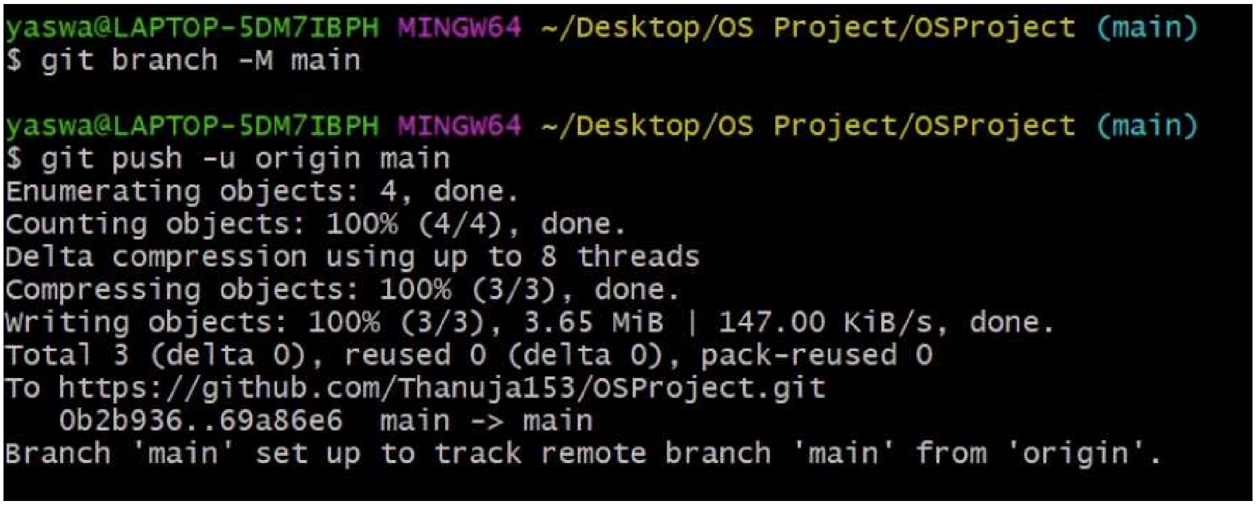
MINGW64:/c/Users/yaswa/Desktop/OS Project/OSProject

```
yaswa@LAPTOP-5DM7IBPH MINGW64 ~/Desktop/OS Project (master)
$ git clone https://github.com/Thanuja153/OSProject.git
Cloning into 'OSProject'...
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 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.03 MiB | 44.00 KiB/s, done.

yaswa@LAPTOP-5DM7IBPH MINGW64 ~/Desktop/OS Project (master)
$ cd OSProject

yaswa@LAPTOP-5DM7IBPH MINGW64 ~/Desktop/OS Project/OSProject (main)
$ git init
Reinitialized existing Git repository in C:/Users/yaswa/Desktop/OS Project/OSProject
/.git/

yaswa@LAPTOP-5DM7IBPH MINGW64 ~/Desktop/OS Project/OSProject (main)
$ git remote --verbose
origin  https://github.com/Thanuja153/OSProject.git (fetch)
origin  https://github.com/Thanuja153/OSProject.git (push)
```

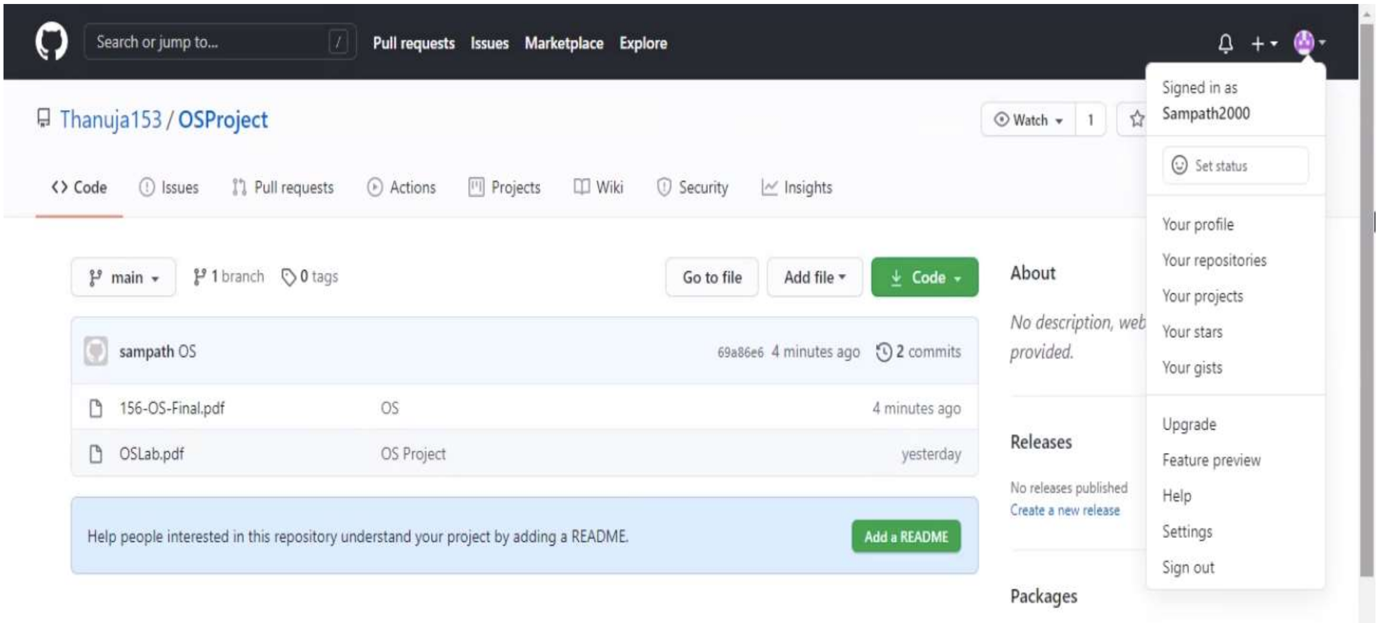


MINGW64:/c/Users/yaswa/Desktop/OS Project/OSProject

```
origin  https://github.com/Thanuja153/OSProject.git (push)

yaswa@LAPTOP-5DM7IBPH MINGW64 ~/Desktop/OS Project/OSProject (main)
$ git add .

yaswa@LAPTOP-5DM7IBPH MINGW64 ~/Desktop/OS Project/OSProject (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:   156-OS-Final.pdf


yaswa@LAPTOP-5DM7IBPH MINGW64 ~/Desktop/OS Project/OSProject (main)
$ git commit -m "OS"
[main 69a86e6] OS
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 156-OS-Final.pdf

yaswa@LAPTOP-5DM7IBPH MINGW64 ~/Desktop/OS Project/OSProject (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```



```
yaswa@LAPTOP-5DM7IBPH MINGW64 ~/Desktop/OS Project/OSProject (main)
$ git branch -M main

yaswa@LAPTOP-5DM7IBPH MINGW64 ~/Desktop/OS Project/OSProject (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 3.65 MiB | 147.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Thanuja153/OSProject.git
   0b2b936..69a86e6  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```



Thanuja153 / OSProject

Signed in as Sampath2000

sampath OS		69a86e6 4 minutes ago 2 commits
156-OS-Final.pdf	OS	4 minutes ago
OSLab.pdf	OS Project	yesterday

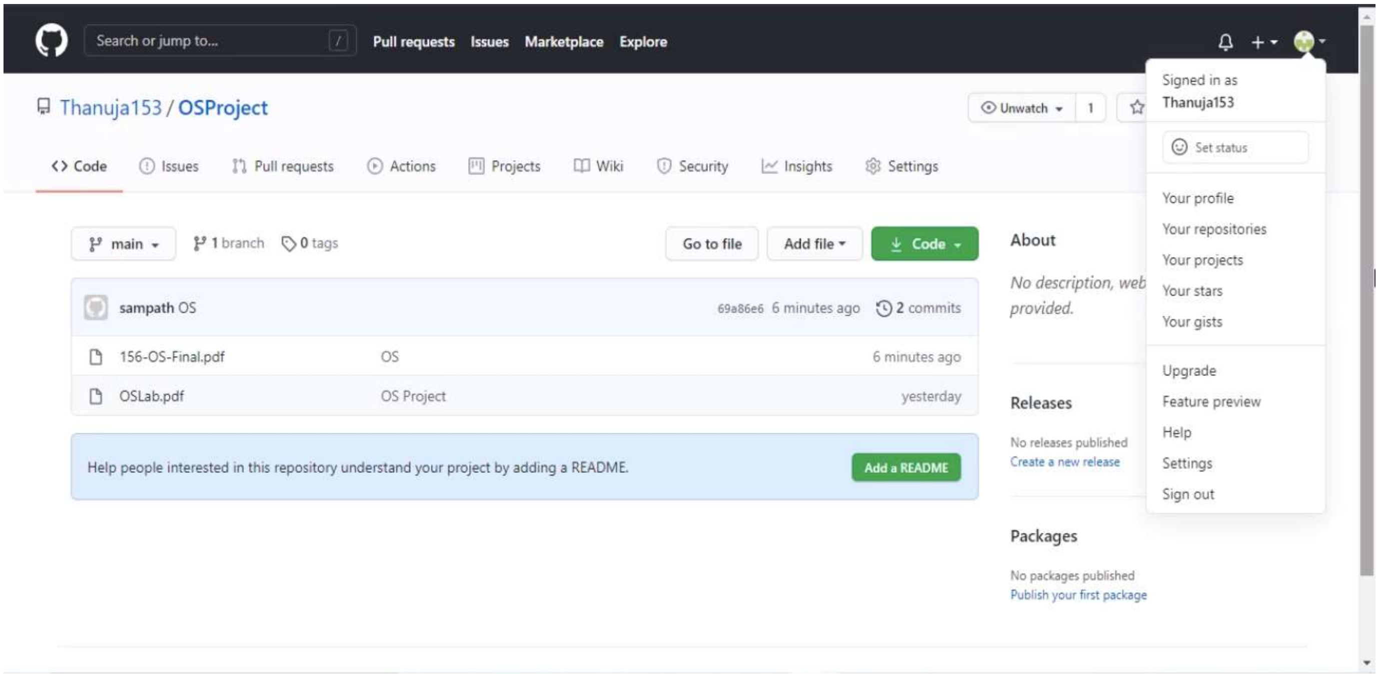
Help people interested in this repository understand your project by adding a README.

No description, web... provided.

No releases published

Here we can see that the developer has cloned the project and updated with some files.

Masters Repository:



sampath OS 69a86e6 6 minutes ago 2 commits

156-OS-Final.pdf	OS	6 minutes ago
OSLab.pdf	OS Project	yesterday

Help people interested in this repository understand your project by adding a README.

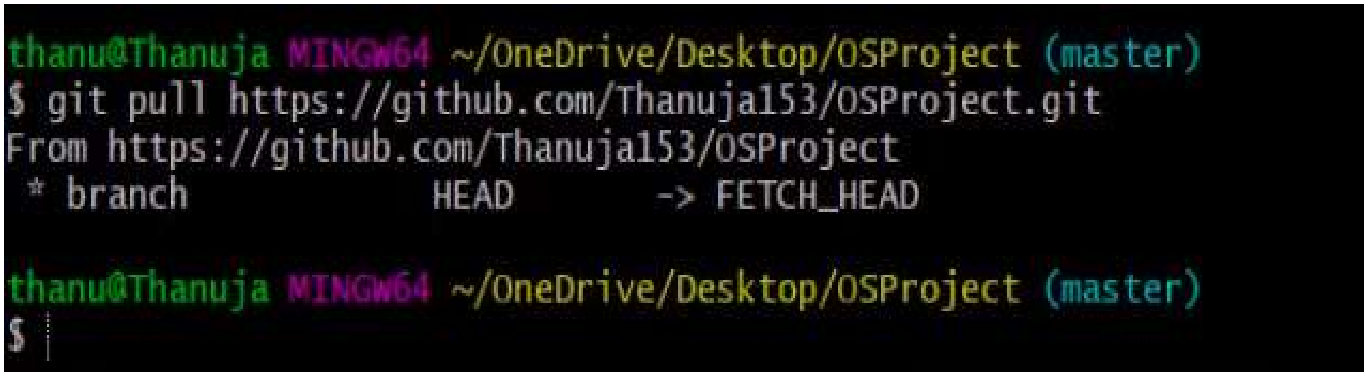
Here we can see that the changes made by some developer has made the masters repository updated.

If the master want to save the changes from remote repository to local repository.

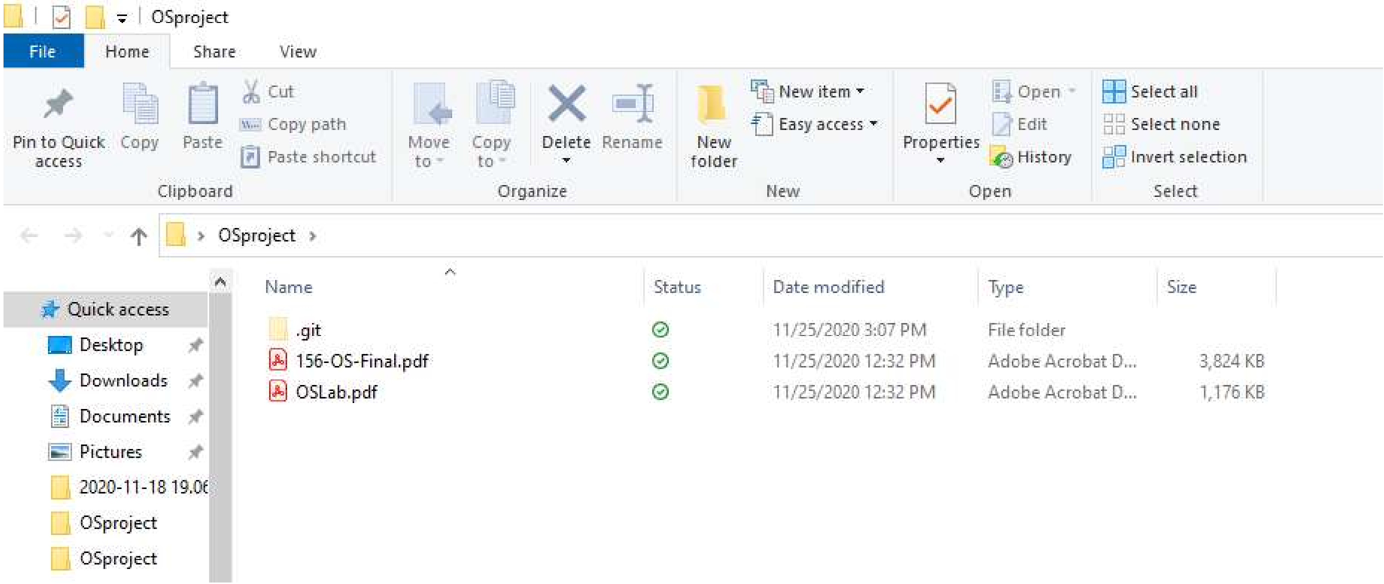
Step1:git pull url(repository)

This command is used to make changes in the local repository by adding the remote repository files to local repository.

Output:



```
thanu@Thanuja MINGW64 ~/OneDrive/Desktop/OSProject (master)
$ git pull https://github.com/Thanuja153/OSProject.git
From https://github.com/Thanuja153/OSProject
 * branch            HEAD       -> FETCH_HEAD

thanu@Thanuja MINGW64 ~/OneDrive/Desktop/OSProject (master)
$ 
```



Name	Status	Date modified	Type	Size
.git	✓	11/25/2020 3:07 PM	File folder	
156-OS-Final.pdf	✓	11/25/2020 12:32 PM	Adobe Acrobat D...	3,824 KB
OSLab.pdf	✓	11/25/2020 12:32 PM	Adobe Acrobat D...	1,176 KB

This shows that we can also update our local repository from the updated remote repository.

Conclusion:

Repository provides storage for a collection of content. This Repository Synchronization can help developers in the same team to be able to view and contribute to the project at the same time without overwriting eachother's work.Here the developers can clone a project and update with some extra features and make the project more efficient and good. After finishing the Repository synchronization, the repository made by the some other users is added to the list of repositories of the master and help them to improve their work.