GIT Assignment

Distributed version control system (DVCS)

1.Create a Distributed version control system, and upload files into it.

Step 1 - On the github page , click on the New Repository button.A form page will appear asking for some information

Step 2 - fill the details and click on the create repository button.

Step 3 - create/upload files with the help of add file button.

Search or jump to...   /   Pull requests   Issues   Codespaces   Marketplace   Explore

🖥 q-madhuri / Assignment   Public

📌 Pin   👁 Unwatch 1 ▾   Fork 0 ▾   ⭐ Star 0 ▾

<> Code   ⊙ Issues   ⨩ Pull requests   ⊙ Actions   ⊞ Projects   📖 Wiki   🛡 Security   📈 Insights   ⚙ Settings

🎋 main ▾   🎋 1 branch   🏷 0 tags   Go to file   Add file ▾   <> Code ▾

Create new file

Upload files

🗂 q-madhuri Add files via upload   ⊙ 3 commits

📄 README.md   Initial commit   3 minutes ago

📄 sample.txt   Create sample.txt   2 minutes ago

📄 test.xlsx   Add files via upload   now

README.md   ✏

Assignment

About

No description, website, or topics provided.   ⚙

📖 Readme

⭐ 0 stars

👁 1 watching

🎋 0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Terms   Privacy   Security   Status   Docs   Contact GitHub   Pricing   API   Training   Blog   About

2.Replicate a DVCS on your local machine, make changes to its files, and then put them back into it.
 Step1 - git clone "https://github.com/madhuri2604/Git-Assignment.git"
Step 2 - make changes in the file sample.txt
Step 3 - git add sample
Step 4 - git commit -m "Modifying the file"
Step 5 - git push origin main

```
91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git (master)
$ git clone https://github.com/madhuri2604/Git-Assignment.git
Cloning into 'Git-Assignment'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), 6.95 KiB | 6.96 MiB/s, done.
Resolving deltas: 100% (1/1), done.

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git (master)
$ ls
Git-Assignment/

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git (master)
$ cd Git-Assignment/

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ ls
README.md  sample  test.xlsx

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   sample

no changes added to commit (use "git add" and/or "git commit -a")

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git add sample

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git commit -m "Modifying the file"
[main 5359154] Modifying the file
 1 file changed, 1 insertion(+)
```
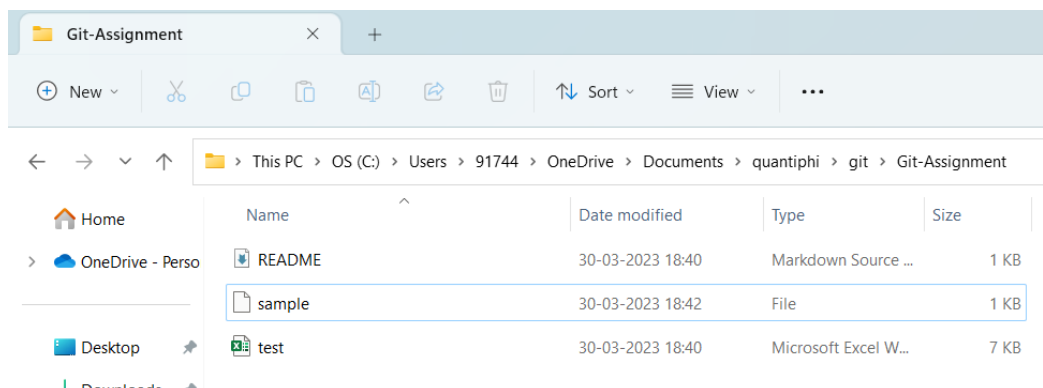
Git-Assignment	×	+

New | Sort | View | ...

This PC > OS (C:) > Users > 91744 > OneDrive > Documents > quantiphi > git > Git-Assignment

Name	Date modified	Type	Size
Home			
OneDrive - Perso	README	30-03-2023 18:40	Markdown Source ...	1 KB
	sample	30-03-2023 18:42	File	1 KB
Desktop	test	30-03-2023 18:40	Microsoft Excel W...	7 KB

MINGW64:/c/Users/91744/OneDrive/Documents/quantiphi/git/Git-Assignment

```
91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git commit -m "Modifying the file"
[main 5359154] Modifying the file
 1 file changed, 1 insertion(+)

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git pull origin main
From https://github.com/madhuri2604/Git-Assignment
 * branch            main       -> FETCH_HEAD
Already up to date.

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 347 bytes | 347.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/madhuri2604/Git-Assignment.git
   6502bbd..5359154  main -> main

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$
```

Search or jump to...   Pull requests   Issues   Codespaces   Marketplace   Explore

madhuri2604 / Git-Assignment   Public

Pin   Unwatch 1   Fork 0

<> Code   Issues   Pull requests   Actions   Projects   Wiki   Security   Insights   Settings

main   1 branch   0 tags

Go to file   Add file   <> Code

madhuri2604 Modifying the file   5359154 4 minutes ago   4 commits

README.md   Initial commit   16 minutes ago
sample   Modifying the file   4 minutes ago
test.xlsx   Add files via upload   15 minutes ago

README.md

Git-Assignment

About
No description, website, or topics provided.

- Readme
- 0 stars
- 1 watching
- 0 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

3. List down all the files which have changed in the last commit. List only the file names.

Step 1 - use git log command to see the commits.
Step 2 - `git show --name-only`

4.You've accidentally added some files on a DVCS. Remove those files from the DVCS without deleting them from your local system.

Step 1- Lets add files in it -(test.xlsx) and check the status.

```
MINGW64:/c/Users/91744/OneDrive/Documents/quantiphi/git/Git-Assignment                    —    □    ×

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git rm --cached test.xlsx
rm 'test.xlsx'

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (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)
        deleted:    test.xlsx

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


91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git commit -m "deleting the test.xlsx file"
[main 34a0310] deleting the test.xlsx file
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 test.xlsx

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git ls-files -c
README.md
sample

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ |
```

both of them and then merge the environments into one

5.You've accidentally committed some files on a DVCS. Revert the DVCS to a previous stable state.

Step 1 : git revert <commit_ID>

The commit_id is visible in logs

```
commit 5359154566678f8bfc7c503d8c56bfdf0791ec7e (origin/main, origin/HEAD)
Author: madhuri2604 <madhurimalviya2601@gmail.com>
Date:   Thu Mar 30 18:42:55 2023 +0530

    Modifying the file

commit 6502bbdba40dd4efced6778793790aa341313d6b
Author: madhuri2604 <64089279+madhuri2604@users.noreply.github.com>
Date:   Thu Mar 30 18:31:51 2023 +0530

    Add files via upload

commit e9f54a2f8b577f4b460ad885b02ab1af332fd42e
Author: madhuri2604 <64089279+madhuri2604@users.noreply.github.com>
Date:   Thu Mar 30 18:31:18 2023 +0530

    Create sample

commit d7dc507312e4ecdafea271f027a624d105c9f771
Author: madhuri2604 <64089279+madhuri2604@users.noreply.github.com>
Date:   Thu Mar 30 18:31:00 2023 +0530

    Initial commit

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git revert ^C

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git revert 5359154566678f8bfc7c503d8c56bfdf0791ec7e
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

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

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

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$
```
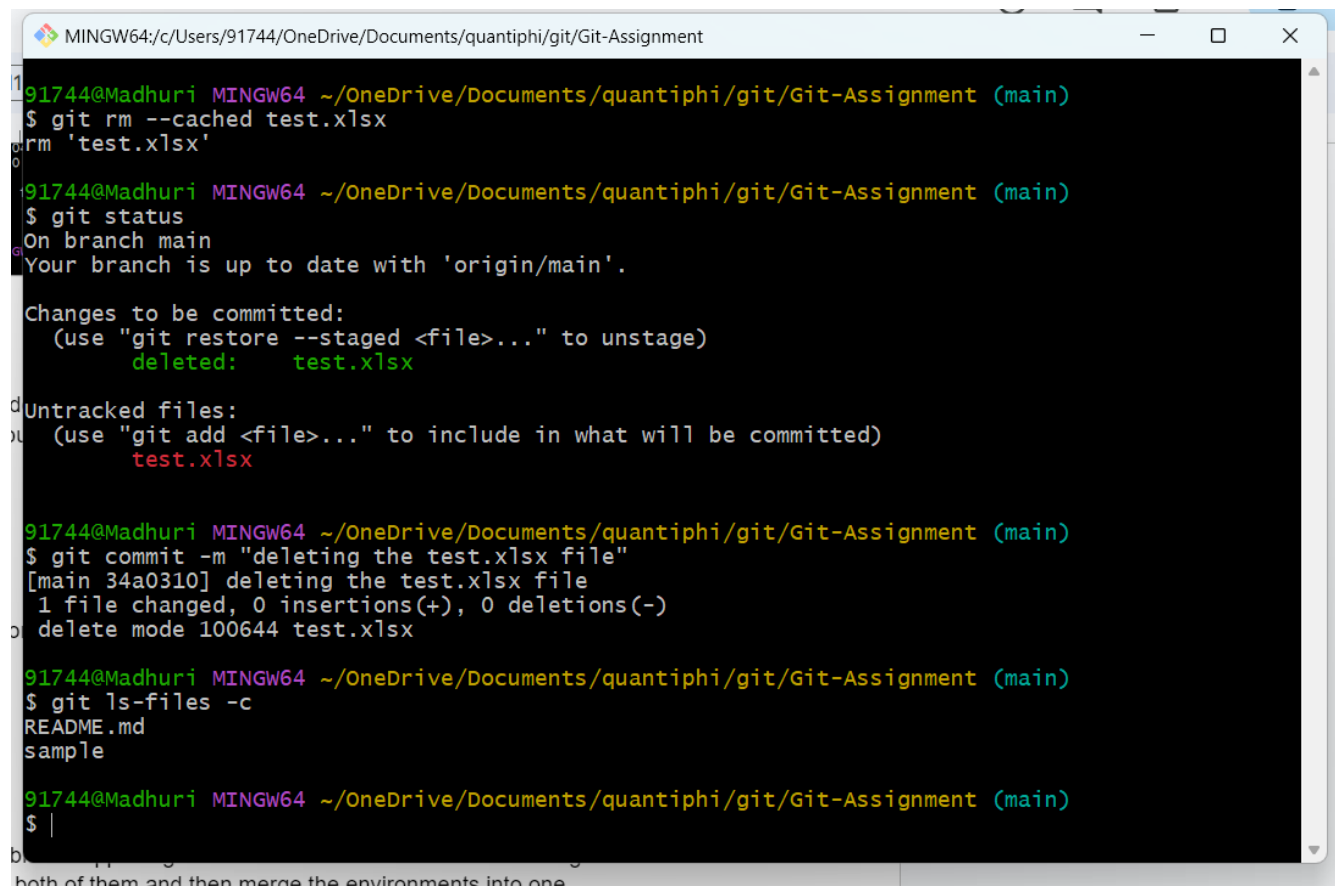
MINGW64:/c/Users/91744/OneDrive/Documents/quantiphi/git/Git-Assignment

```
91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git log
commit 7231cd535641bcd6575277a0d10f1c9791693244 (HEAD -> main)
Author: madhuri2604 <madhurimalviya2601@gmail.com>
Date:   Thu Mar 30 19:17:45 2023 +0530

    Revert "Modifying the file"

    This reverts commit 5359154566678f8bfc7c503d8c56bfdf0791ec7e.
```

6.Create a DVCS capable of supporting 2 similar dev environments. Make changes in the files which are present on both of them and then merge the environments into one.

Step 1 - create second environment(branch)
Step 2 - make some changes to the sample file from newbranch environment & commit it

Step 3 - go to main branch & modify, commit the sample file there
Step 4 - merge the branches

Git merge <branch-name>

```
    (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
        modified:   sample

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

no changes added to commit (use "git add" and/or "git commit -a")

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git add sample

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git commit -m "modifying from main"
[main b0aaa45] modifying from main
 1 file changed, 2 insertions(+)

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git diff main newbranch
diff --git a/sample b/sample
index e256c76..2feba1c 100644
--- a/sample
+++ b/sample
@@ -1,3 +1,3 @@
-
-new change done by main branch
 this is a new file
+making some changes
+

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git merge newbranch
Auto-merging sample
Merge made by the 'ort' strategy.
 sample | 2 ++
 1 file changed, 2 insertions(+)

91744@Madhuri MINGW64 ~/OneDrive/Documents/quantiphi/git/Git-Assignment (main)
$ git log --oneline --graph --decorate
*   145e99c (HEAD -> main) Merge branch 'newbranch'
|\
| * 18e382e (newbranch) changing from newbranch
* | b0aaa45 modifying from main
|/
* 7231cd5 Revert "Modifying the file"
* 34a0310 deleting the test.xlsx file
* 5359154 (origin/main, origin/HEAD) Modifying the file
* 6502bbd Add files via upload
```

7.You have a DVCS supporting your prod, test and dev environments, you need to make sure that only a particular set of commits from the test environment are added in the prod environment. What & how would you do it?

Step 1 - create 3 env = prod, dev, test

Step 2 - in test env , add and commit files.

Step 3 - now go to prod env and use cherry-pick command to select the commits you want to add to prod env

```
git cherry-pick <commit-ids>
```

Step 4- we only added 1st commit from test env therefore prod env does not contain file2.txt

Scenario Questions:

1.If the central repository is two commits ahead of your local repository, how will you push your code ?

First we need to pull the changes(`git pull`) from the remote repository to local repository and resolve any conflicts that may arise during the merge ,

`git commit`

Commit the changes to the local repository.

and then push the code `git push`

2.I have two branches, A and B, where A is my main branch, and B is a the branch with new feature, describe the step by step process getting all of B's code in A

Step 1 - In B branch , stage and commit the files

Step 2 - checkout to branch A

Step 3 - merge branch B with A

`git  merge  B`

All the changes in B branch will merge into A

3.Say you have to do an experimental change but still want to have a clean copy of your old code, should you use branching ? Explain your decision

Yes definitely , branching is the best choice to keep a clean copy of your code.Branching allows you to create a separate copy of your code where you can try and experiment on without affecting the main codebase.

It also helps in collaborating with others.